Spatio-Temporal Knowledge Transfer for Urban Crowd Flow Prediction via Deep Attentive Adaptation Networks

Senzhang Wang, *Member, IEEE*, Hao Miao, Jiyue Li, and Jiannong Cao, *Fellow, IEEE*

***Abstract*—Accurately predicting the urban spatio-temporal data is critically important to various urban computing tasks for smart city related applications such as crowd flow prediction and traffic congestion prediction. Existing models especially deep learning based approaches require a large volume of training data, whose performance may degrade remarkably when the data is scarce. Recent works try to transfer knowledge from the intra-city or cross-city multi-modal spatio-temporal data. However, the careful design of what to transfer and how between the multi-modal spatio-temporal data needs to be determined in advance. There still lacks an end-to-end solution that can automatically capture the common cross-domain knowledge. In this paper, we propose a Deep Attentive Adaptation Network model named ST-DAAN to transfer cross-domain Spatio-Temporal knowledge for urban crowd flow prediction. ST-DAAN first maps the raw spatio-temporal data of source domain and target domain to a common embedding space. Then domain adaptation is adopted on several domain-specific layers through adding a domain discrepancy penalty to explicitly match the mean embeddings of the two domain distributions. Considering the complex spatial correlation in many urban spatio-temporal data, a global attention mechanism is also designed to enable the model to capture broader spatial dependencies. Using urban crowd flow prediction as a demonstration, we conduct experiments on five real-world large datasets over both intra- and cross-city transfer learning. The results demonstrate that ST-DAAN outperforms state-of-the-art methods by a large margin.**

***Index Terms*—Spatio-temporal data, transfer learning, urban computing, crowd flow prediction.**

I. Introduction

THE spatio-temporal data generated in urban areas are ubiquitous nowadays with the broad application of various position techniques such as Global Position System (GPS), mobile devices and remote sensing. Accurately predicting the urban spatio-temporal data is practically useful to support many urban computing tasks for the construction of smart cities including human mobility mining [1], traffic prediction [2] and urban planning [3]. However, it is very difficult for traditional statistics-based approaches such as linear regression to achieve desirable prediction performance on urban spatio-temporal data [4] due to the following two reasons. First, the urban spatio-temporal data are multi-modal containing taxi trajectories, bike trajectories, metro/bus swiping data, *et al.* These data are usually highly correlated, and thus should be analyzed jointly rather than separately. Second, these data present very complex spatio-temporal dependencies, which are hard to be captured by shallow models.

Recently deep learning has enjoyed considerable success in many domains, and also achieved remarkable performance gains in various spatio-temporal prediction tasks [1], [4]–[6]. [1] proposed a deep learning model ST-ResNet to collectively forecast the inflow and outflow of crowds in each region of a city. [7] proposed a Spatio-Temporal Dynamic Network (STDN) model for road network based traffic prediction. [4] proposed to use the attention-based neural network which combined encoder-decoder framework and ConvLSTM to predict the passenger pickup/dropoff demands for the mobility-on-demand services. [8] proposed the DeepTransport model which combined CNN and RNN to predict the traffic data within a transport network. However, deep learning based methods usually require a large number of training data. In real applications, however, the spatio-temporal data can be scarce due to various reasons such as data collection mechanisms (e.g. low sampling rate), data privacy issues and low city development level [9]. Their performance may degrade remarkably with insufficient training data. Additionally, these models are specially designed for one particular data type, and are hard to be generalized to handle other types of spatio-temporal data.

Some recent efforts have been made to use transfer learning techniques to address the data scarcity issue via using the rich spatio-temporal data from other sources [9], [10]. However, the work [9] needs other service data to help identify where of a city transfer learning can be performed in first, and then uses it as constraint adding to a deep learning based prediction model. [10] focuses on capturing the common long-term trend (i.e., periodicity) of the spatio-temporal data from multiple cities, and then transferring it to the target city. There still lacks an end-to-end model to automatically perform

Manuscript received April 7, 2020; revised September 11, 2020 and November 18, 2020; accepted December 3, 2020. Date of publication February 10, 2021; date of current version May 3, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1003900, in part by the HK RGC Collaborative Research Fund under Grant C5026-18G, in part by the HK RGC Research Impact Fund under Grant R5034-18, in part by the CCF-Tencent Open Research Fund, and in part by the Fundamental Research Funds for the Central Universities under Grant NZ2020014. The Associate Editor for this article was Y. Lv. *(Corresponding author: Senzhang Wang.)*

Senzhang Wang is with the School of Computer Science and Engineering, Central South University, Changsha 410083, China.

Hao Miao and Jiyue Li are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: senzhangwang@gmail.com).

Jiannong Cao is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong.

Digital Object Identifier 10.1109/TITS.2021.3055207

spatio-temporal knowledge transfer and predictive learning in a unified learning framework.

In this paper, we correlate the cross-domain urban spatio-temporal data from the data distribution perspective rather than the region-level data sample perspective [9], and propose a Deep Attentive Adaptation Network named ST-DAAN to perform spatio-temporal knowledge transfer and predictive learning. Deep adaptation network (DAN) [11] generalizes deep convolutional neural network to the domain adaptation scenario through learning transferable latent features between source domain and target domain. Inspired by the success of DAN in various transfer learning tasks in computer vision, we borrow its idea to learn the transferable features of spatio-temporal data through matching the domain distributions in the latent feature space. Specially, ST-DAAN first maps the raw spatio-temporal data of two domains to a common embedding space by several ConvLSTM layers. The embedded features are then input into domain-specific 3D convolution layers to learn domain-specific features. The domain-specific features are then embedded to a reproducing kernel Hilbert space for matching the embedding distributions of the two domains. To reduce the domain discrepancy, a penalty on maximum mean discrepancies (MMD), which is widely used as distribution distance measure, is added to each domain-specific layer for joint optimization. Finally, a global spatial attention mechanism is also designed to more broadly capture the spatial dependencies of the data in two domains.

The major contributions of the work are as follows.

- A novel deep learning framework is presented to perform spatio-temporal knowledge transfer for urban crowd flow prediction in an end-to-end learning way. To our knowledge, this is the first end-to-end transfer learning framework for cross-domain urban crowd flow prediction.
- Deep adaption network is utilized to perform knowledge transfer in domain-specific feature learning layers. DAN confines the domain discrepancy through adding the penalty on the maximum mean discrepancies between their embedding distributions, which to our knowledge is the first work that performs urban spatio-temporal knowledge transfer from the perspective of embedding domain distribution.
- A global spatial attention mechanism is designed to enable the model to capture the spatio-temporal data correlations in a broader area (the entire city), which is especially useful in cross-city transfer learning. Such correlations are incorporated into the final prediction model to improve the prediction accuracy.

The remainder of the paper is organized as follows. We first discuss related works in Section 2, and then formally define the studied problem in Section 3. Section 4 introduces the proposed model ST-DAAN and gives an algorithm to effectively solve the model. In Section 5, we evaluate our approach and report the results. Finally, we conclude the work in Section 6.

II. Related Work

This work is highly relevant to the research topics of urban crowd flow prediction, transfer learning and deep adaptation network (DAN). Next we will review related works from the three aspects.

A. Urban Crowd Flow Prediction

In recent years, sptiao-temporal data prediction has attracted rising research interest due to the increasingly available urban data and the rich applications. Traditionally, statistics-based time series models such as ARIMA [12], [13] and regression models with spatio-temporal regularization [14] are used for spatio-temporal prediction. Due the limited learning capacity, such statistics-based methods usually cannot capture the very complex spatio-temporal dependencies of the urban data. Recently, various deep learning methods are broadly applied and have achieved much better performance than traditional statistics-based shallow models, such as DNN [15], ST-ResNet [1], SeqST-GAN [16], ConvLSTM [17] and MT-ASTN [18]. [1] proposed to use residual CNN on the images of traffic flow. These methods simply used CNN on the whole city and all the regions for prediction. [7] proposed a Spatio-Temporal Dynamic Network (STDN) model for road network based traffic prediction. STDN combined CNN model and RNN model to capture both spatial and temporal correlations. [4] was the first recent work that studied the problem of multi-step taxi passenger demand prediction. [4] proposed to use the attention-based neural network which combined encoder-decoder framework and ConvLSTM to predict the passenger pickup/dropoff demands for the mobility-on-demand services. [19] proposed the Diffusion Convolutional Recurrent Neural Network (DCRNN) to model the traffic flow as a diffusion process on a directed road graph, which is a deep learning framework for traffic flow forecasting. [20] proposed a multi-task deep learning framework to simultaneously predict the node flow and edge flow in a constructed urban spatio-temporal network. [21] proposed a model entitled UrbanFM to infer the real-time and fine-grained crowd flows throughout a city based on coarse-grained observations. [22] proposed a novel framework that employed matrix factorization for spatio-temporal neural networks to improve the performance of urban flow prediction. [23] provided a survey on urban flow prediction from spatiotemporal data using machine learning. It summarized recent works and methodologies of urban urban flow prediction, and showed open challenges of this research topic. [24] reviewed recent advances in applying deep learning techniques for various spatio-temporal data mining tasks. However, these models are mostly supervised, and a large number of training data are required. Scarce training data will lead to overfitting due to the large number of parameters in deep learning models.

To address the data scarcity issue, several recent works tried to use transfer learning techniques to address the cross-city spatio-temporal data prediction task. [25] provided a general urban transfer learning paradigm, which summarized the common transfer strategies to take, general steps to follow, and case studies in various urban computing tasks including public safety and transportation management. [9] proposed a cross-city transfer learning method called *RegionTrans* for urban spatio-temporal data prediction. *RegionTrans* transferred knowledge from a data-rich source city to a data-scarce target city. An inter-city region matching function was first learned to match the regions between two cities where knowledge can be transferred. [10] leveraged the spatio-temporal data from multiple source cities to increase the stability of transferring knowledge to the target city. A meta-learning model named *MetaST* is proposed in [10] to learn well-generalized initialization of the prediction model parameters, which can be adapted to the target model. *MetaST* focused on transferring the long-term spatio-temporal patterns of different cities, but the short-term or real-time features such as periodicity cannot be effectively captured and leveraged.

B. Transfer Learning

Transfer learning focuses on addressing the scarce labeled data problem in machine learning. [26] and [27] both gave comprehensive surveys on summarizing the categories and existing methods for transfer learning. [28] proposed the TCA model which learned some transfer components across domains in a reproducing kernel Hilbert space using maximum mean discrepancy. TLDA [29] was proposed which adopted two autoencoders for the source domain and target domain that shared the same parameters. [30] proposed Joint Adaptation Networks (JAN), which learned a transfer network by aligning the joint distributions of multiple domain-specific layers across domains based on a joint maximum mean discrepancy (JMMD) criterion. [31] utilized matrix tri-factorization to discover both the implicit and the explicit similarities for cross-domain recommendation. In spatio-temporal prediction area, data sparsity problem is often existing when targeted service is new. To overcome this problem, several works try to find a source city with adequate data to transfer knowledge. [32] designed a cross-city transfer learning framework with collaborative filtering and AutoEncoder to conduct chain store site recommendation.

C. Deep Adaptation Network

Domain adaptation is a type of transfer learning method in computer vision that aims to address the lack of massive amounts of labeled data [33]. One of the main approaches to establishing knowledge transfer is to learn domain-invariant models from data, which can bridge the source and target domains in an isomorphic latent feature space. Following this idea, Deep adaptation network (DAN) was proposed in [11], which generalized deep convolutional neural network to the domain adaptation scenario to learn more transferable feature representations in the latent embedding space. DAN has been proven to be very effective in many transfer learning tasks such as person re-identification [34], object detection [35], and disease diagnostics [36]. Inspired by the success of DAN in computer vision, in this paper we for the first time adopt the idea of DAN for transferring cross-domain spatio-temporal knowledge for urban crowd flow prediction.

III. Definition and Problem Formulation

In this section, we will first define some terminologies to help us state the studied problem. Then a formal problem definition will be given.

Definition 1 (Cell Region): We partition a city c into a $m \times n$ grid map based on the longitude and latitude. Each grid is defined as a cell region, and all the grids form a cell region set $R_c = \{r_{1,1}, \ldots r_{i,j}, \ldots r_{m,n}\}$, where $r_{i,j}$ is the cell region in the $i\text{-}th$ row and $j\text{-}th$ column of the grid map.

Definition 2 (Spatio-Temporal Image): Given the cell regions $R = \{r_{1,1}, \ldots r_{i,j}, \ldots r_{m,n}\}$ of a city and the spatio-temporal measurement (e.g. traffic volume, crowd flow, etc.) in each cell region, we consider the spatio-temporal measurement in time-stamp t in all the regions as a spatio-temporal image denoted as a matrix $\mathbf{X}_t \in \mathcal{R}^{m \times n}$.

Definition 3 (Spatio-Temporal Image Time Series): Given the current time-stamp t and the length of the time-stamps k, we denote the spatio-temporal image time series of a city as: $\mathcal{X}_t = \{\mathbf{X}_{t-k+1}, \ldots \mathbf{X}_t\} \in \mathcal{R}^{m \times n \times k}$. Note that the spatio-temporal image time series $\mathcal{X}_t$ is a 3-dimensional tensor, whose entry $\mathcal{X}(i, j, t)$ denotes the spatio-temporal measurement in time-stamp t and cell region $r_{i,j}$.

Based on the above terminology definitions, we formally define the studied problem as follows.

Problem Definition 1 (Spatio-Temporal Prediction With Knowledge Transfer): Given the rich spatio-temporal image time series $\mathcal{X}^{\mathcal{S}} = \{\mathcal{X}_t^{\mathcal{S}} | 0 \leq t \leq L^{\mathcal{S}}\}$ in the source domain $\mathcal{S}$, and the scarce spatio-temporal image time series $\mathcal{X}^{\mathcal{T}} = \{\mathcal{X}_t^{\mathcal{T}} | 0 \leq t \leq L^{\mathcal{T}}\}$ in the target domain $\mathcal{T}$, we aim to learn a function f to predict the spatio-temporal image of the target domain in the next time-stamp by transferring the knowledge from the source domain:

$$\begin{aligned} &\min_f \sum_t \text{error}\left(\mathbf{Y}_t^{\mathcal{T}}, \tilde{\mathbf{Y}}_t^{\mathcal{T}}\right) \\ &\text{where } \tilde{\mathbf{Y}}_t^{\mathcal{T}} = f\left(\mathcal{X}_t^{S}, \mathcal{X}_t^{\mathcal{T}}\right) \end{aligned} \tag{1}$$

where $\mathcal{S}$ denotes the source domain and $\mathcal{T}$ denotes the target domain. $L^{\mathcal{S}}$ and $L^{\mathcal{T}}$ denote the data size of the two domains and $L^{\mathcal{T}} \ll L^{\mathcal{S}}$. $\tilde{\mathbf{Y}}_t^{\mathcal{T}}$ is the predicted spatio-temporal image after $\mathcal{X}_t^{\mathcal{T}}$, and $\mathbf{Y}_t^{\mathcal{T}}$ is the ground truth. $error$ is the prediction error which can be mean absolute error, root mean squared error, etc.

Note that the spatio-temporal data of two domains can be collected from the same city or two different cities. If the target domain data comes from a different city from the source domain, we require that the two cites are divided into the same size of cell regions.

IV. Deep Attentive Adaptation Network

Fig. 1 shows the framework of the proposed ST-DAAN. As shown in the figure, the model contains three major steps. First, several ConvLSTM layers that are shared by both source and target domains are stacked to learn the hidden representation of the raw input spatio-temporal data. This step aims to capture the spatio-temporal dependencies of the data and map the data into a common embedding latent space. Next, a deep adaptation network structure that is instantiated with several domain-specific 3D convolutional layers are applied for further data representation learning. This step performs spatio-temporal knowledge transfer on the domain-specific layers by

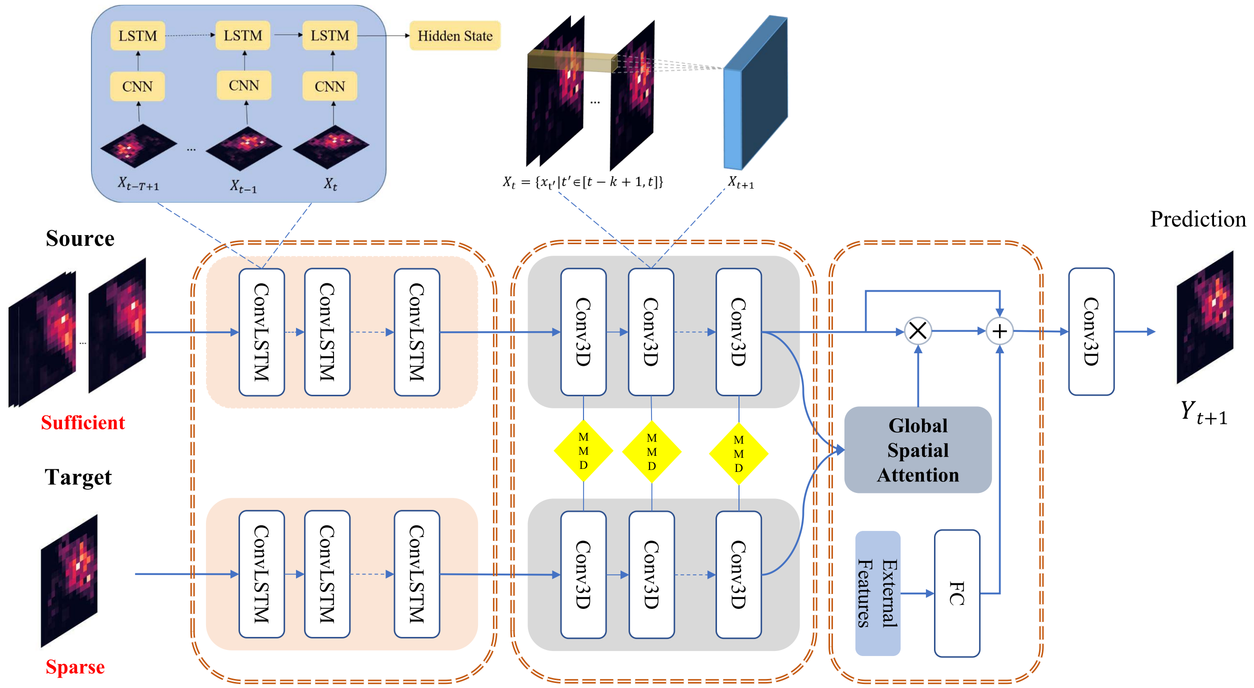


Fig. 1. The model framework. It contains three major steps. First, several ConvLSTM layers (the left part of the figure) that are shared by both source and target domains are stacked to learn the hidden representation of the raw input spatio-temporal data. Next, a deep adaptation network structure that is instantiated with 3D convolutional layers (the middle part of the figure) is constructed for further data representation learning. Note that the hidden representations of these layers are embedded to a reproducing kernel Hilbert space for knowledge transfer through maximum mean discrepancies (MMD). Finally, a global spatial attention mechanism (the right part of the figure) is designed to capture the global spatial dependencies between each cell region in source domain to all the cell regions in target domain. External features including holidays and weather are also input into a fully connected layer for feature learning.

explicitly reducing the domain discrepancy. To achieve this goal, the hidden representations of these layers are embedded to a reproducing kernel Hilbert space through maximum mean discrepancies (MMD), where the mean embeddings of two domain distributions can be explicitly matched. Finally, a global spatial attention mechanism is designed to capture the global spatial dependencies between each cell region in source domain to all the cell regions in target domain. Next we will elaborate the three parts in detail.

A. Spatio-Temporal Data Representation Learning With ConvLSTM

We first use several stacked ConvLSTM layers to learn the data representations of the raw spatio-temporal image time series. CNN [37] is capable to capture the local spatial correlations for image data, while LSTM [38] is good at learning the temporal correlation. To process spatio-temporal data, the combination of CNN and LSTM, which is called ConvLSTM is proposed [17] and widely used in various spatio-temporal prediction tasks, such as traffic accident prediction [39], crowd flow prediction [40], and precipitation prediction [41]. Thus in this paper we also use ConvLSTM for spatio-temporal image time series representation learning to encode the spatio-temporal dependencies. Note that the ConvLSTM layers are shared by the data of both domains, and thus the data are embedded to a common latent representation space.

Different from the fully-connected LSTM, the input and hidden state of ConvLSTM in a time-stamp are all 3D tensors, and the convolution operation is conducted for both input-to-state and state-to-state connection. More specifically, ConvLSTM does the convolution operation on the data in each time-stamp (i.e. $\mathbf{X}_t$) first, and then pass them along the time span $[t-k+1, \ldots, t]$ through the LSTM module, which can be formulated as:

$$\begin{aligned}
i_t &= \sigma\left(W_{xi} * \mathbf{X}_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i\right),\\
f_t &= \sigma\left(W_{xf} * \mathbf{X}_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f\right),\\
C_t &= f_t \circ C_{t-1} + i_t \circ tanh\left(W_{xc} * \mathbf{X}_t + W_{hc} * H_{t-1} + b_c\right),\\
o_t &= \sigma\left(W_{xo} * \mathbf{X}_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o\right),\\
H_t &= o_t \circ tanh\left(C_t\right),
\end{aligned}$$

where '$*$' denotes the convolution operator, '$\circ$' denotes the Hadamard product, σ is the logistic sigmoid function, i_t, f_t, C_t o_t, and H_t are input gate, forget gate, memory cell, output gate and hidden state, and $W_{\alpha\beta}$ $(\alpha \in \{x, h, c\}, \beta \in \{i, f, o, c\})$ are the parameters of convolutional layers in ConvLSTM. For better illustration, we visualize the structure of convolutional LSTM at the top left of Fig. 1. Several ConvLSTM layers are stacked, and the output of i-th layer can be formulated as follows:

$$\begin{aligned}
\boldsymbol{H}_i^{\mathcal{S}} &= ConLSTM_i\left(\mathcal{X}^{\mathcal{S}}, w_{\mathcal{S}}\right),\\
\boldsymbol{H}_i^{\mathcal{T}} &= ConLSTM_i\left(\mathcal{X}^{\mathcal{T}}, w_{\mathcal{T}}\right),
\end{aligned} \tag{2}$$

where $\boldsymbol{H}_i^{\mathcal{S}}$ and $\boldsymbol{H}_i^{\mathcal{T}}$ are data representations of the two domains learned by the i-th layer $ConvLSTM_i$. The final representations learned through stacked $ConvLSTM_i$ are denoted as $\boldsymbol{H}^{\mathcal{S}}$ and $\boldsymbol{H}^{\mathcal{T}}$.

B. Spatio-Temporal Knowledge Transfer With Deep Adaptation Network

The output $\boldsymbol{H}^{\mathcal{S}}$ and $\boldsymbol{H}^{\mathcal{T}}$ of ConvLSTM are then input into the Deep Adaptation Network (DAN), which is instantiated as stacked $Conv3D$ layers with Maximum Mean Discrepancy constraint for domain-specific feature learning and knowledge transfer. To learn the transferable features between two domains, DAN explicitly reduces the domain discrepancy by embedding the data representations of the domain-specific layers to a reproducing kernel Hilbert space where the mean embeddings of the two domain distributions can be explicitly matched [11].

To match the mean embeddings of the data distributions, we use Maximum Mean Discrepancy (MMD) to measure the distance of the domain distributions in the embedding space. MMD is initially invented for addressing the two-sample test problem. Given two sets of data samples, MMD can determine whether they are from the same distribution through constructing statistical tests [42]. It they are from different distributions, MMD can be also used to measure how different the two distributions is. In our studied problem, we can transfer spatio-temporal knowledge between two domains via adding the constraint that the MMD of the embedded two domain data should be small if they are more transferable. MMD can be calculated as follows:

$$MMD(\mathcal{D}^{\mathcal{S}}, \mathcal{D}^{\mathcal{T}}) = \left\| \frac{1}{n}\sum_{i=1}^{n}\phi\left(d_i^{\mathcal{S}}\right) - \frac{1}{m}\sum_{j=1}^{m}\phi\left(d_j^{\mathcal{T}}\right)\right\|_{\mathcal{H}}^{2}, \tag{3}$$

where $\phi(\cdot)$ denotes the kernel function (here we use Gaussian kernel), and $\mathcal{H}$ denotes the Hilbert space. $\mathcal{D}^{\mathcal{S}} = d_*^{\mathcal{S}}$ and $\mathcal{D}^{\mathcal{T}} = d_*^{\mathcal{T}}$ are the data samples of two domains, respectively.

As the output of the ConvLSTM is 4-dimensional tensors, here we use $Conv3D$ to perform the convolution operation on the tensors to learn the domain-specific data representations $\boldsymbol{F}^{\mathcal{S}}$ and $\boldsymbol{F}^{\mathcal{T}}$, and then calculate MMD loss through Eq.3. The i-th layer can be formulated as Eq.4. We use the same $Conv3D_i$ structure as in the work [43], and one can refer to it for more details.

$$\begin{aligned} \boldsymbol{F}_i^{S} &= Conv3D_i\left(\boldsymbol{H}^{S}, w_{\mathcal{HS}}\right), \\ \boldsymbol{F}_i^{\mathcal{T}} &= Conv3D_i\left(\boldsymbol{H}^{\mathcal{T}}, w_{\mathcal{HS}}\right), \\ mmd\_loss_i &= MMD(\boldsymbol{F}_i^{S}, \boldsymbol{F}_i^{\mathcal{T}}). \end{aligned} \tag{4}$$

C. Global Spatial Attention

The local spatial correlations captured by the convolution operation of CNN usually cannot fully reflect the geographical dependencies in many real application scenarios. Previous works [44], [45] showed that two regions with similar POI distribution or functionality, even though they are not geographically close to each other, can present very similar patterns of the spatio-temporal data (e.g. taxi trips and crowd flow). To more broadly capture the spatial dependency, we propose a global spatial attention mechanism which measures the coefficient between the spatio-temporal data in each region of target domain and all the cell regions in source domain. Fig. 2 shows the illustration of the global spatial attention, where the left matrix is the embedded latent data representation of the source domain and the right matrix is the latent data representation of the target domain. Note that we let the representation matrix the same size ($m \times n$) as the raw input spatio-temporal image, and thus each entry of the matrix is corresponding to the latent representation of the cell region.

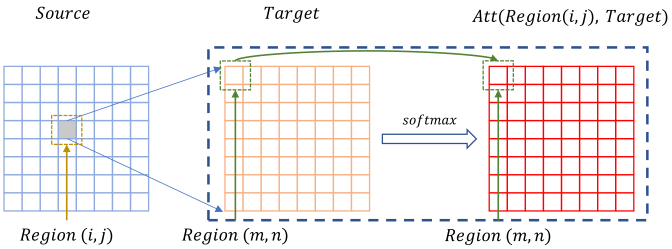


Fig. 2. An illustration of global spatial attention. The left part shows the region features of the source domain, the middle part shows the region features of the target domain, and the right part shows the attention value between $Region(i, j)$ of the source domain and all the regions of the target domain. $Att(Region(i, j), Target)$ is calculated based on the feature similarity between $Region(i, j)$ and the region of the target domain with a $softmax$ function.

Specially, the representation of each cell region in source domain is linearly represented by the weighted sum of all the region representations in target domain. First, we feed the data representations of source and target domains into 3D convolutional layers to generate an embedded representation tensors, and then reshape them into a 2D matrices as follows:

$$\begin{aligned} \boldsymbol{F}_{att}^{\mathcal{S}} &= Conv3D\left(\boldsymbol{F}^{\mathcal{S}}, w_{att}\right), \\ \boldsymbol{F}_{att}^{\mathcal{T}} &= Conv3D\left(\boldsymbol{F}^{\mathcal{T}}, w_{att}\right), \\ \boldsymbol{F}_{att}^{\mathcal{S}} &: R^{1\times C\times m\times n} \rightarrow R^{C\times N}, \\ \boldsymbol{F}_{att}^{\mathcal{T}} &: R^{1\times C\times m\times n} \rightarrow R^{C\times N}, \end{aligned} \tag{5}$$

where N is equal to $m \times n$, and $C = 2$ represents inflow and outflow. Each column of $\boldsymbol{F}_{att}^{\mathcal{S}}$ and $\boldsymbol{F}_{att}^{\mathcal{T}}$ represents the representation of a region. The attention matrix $\mathcal{A}$ is calculated by first conducting the dot-product operation between transposed $\boldsymbol{F}_{att}^{\mathcal{S}'} \in \mathbb{R}^{N\times C}$ and $\boldsymbol{F}_{att}^{\mathcal{T}} \in \mathbb{R}^{C\times N}$, and then performing the $Softmax$ operation as follows:

$$\mathcal{A} = \text{Softmax}\left(\boldsymbol{F}_{att}^{\mathcal{S}'} \otimes \boldsymbol{F}_{att}^{\mathcal{T}}\right). \tag{6}$$

With the attention matrix, we can compute the correlation weight of each region in target domain to all the regions in source domain by summing the features of all regions with the calculated attention weights. We implement this process with a dot-product operation. First, we reshape $\boldsymbol{F}^{\mathcal{S}}$ into a 2D matrix, and then conduct dot-product between $\boldsymbol{F}^{\mathcal{S}}$ and $\mathcal{A}$ to compute the global spatial feature $\boldsymbol{F}_{\mathcal{S}}^{g}$. Finally $\boldsymbol{F}_{\mathcal{S}}^{g}$ is further reshaped to dimension $l \times C \times m \times n$. The entire process can be expressed as:

$$\begin{aligned} \boldsymbol{F}^{\mathcal{S}} &: \mathbb{R}^{l\times C\times m\times n} \rightarrow \mathbb{R}^{C\times N} \\ \boldsymbol{F}_{g}^{\mathcal{S}} &= \boldsymbol{F}^{\mathcal{S}} \otimes \mathcal{A} \\ \boldsymbol{F}_{g}^{\mathcal{S}} &: \mathbb{R}^{C\times N} \rightarrow \mathbb{R}^{l\times C\times m\times n}. \end{aligned} \tag{7}$$

Algorithm 1 Deep Attentive Adaptation Network

```
Input: $\mathcal{X}^{\mathcal{S}}$: source domain data; $\mathcal{X}^{\mathcal{T}}$ : target domain data;
Output: parameter set $\Theta$
initialize parameters $\Theta$
epoch $\leftarrow 0$
$N \leftarrow \lfloor \frac{L^{\mathcal{S}}}{L^{\mathcal{T}}} \rfloor$
while $not\ converge$ do
    $mmd\_loss \leftarrow 0$
    $\mathcal{L} \leftarrow 0$, $\mathcal{L}$ is $\mathcal{L}$-th time stamp
    for $t \in L^{\mathcal{T}}$ do
        sample $\{\mathcal{X}_t^{\mathcal{T}}, \mathbf{Y}_t^{\mathcal{T}}\} \in \mathcal{X}^{\mathcal{T}}$
        $\boldsymbol{H}^{\mathcal{T}} \leftarrow$ feature learning with $\mathcal{X}_t^{\mathcal{T}}$ by Eq.2
    end for
    while $t \le L^{\mathcal{S}}$ do
        if $\mathcal{L} \le N$ then
            sample $\{\mathcal{X}_t^{\mathcal{S}}, \mathbf{Y}_t^{\mathcal{S}} | t \in [t, t+L^{\mathcal{T}})\} \in \mathcal{X}^{\mathcal{S}}$
            $\boldsymbol{H}^{\mathcal{S}} \leftarrow$ feature learning with $\mathcal{X}_t^{\mathcal{S}}$ by Eq.2
            $mmd\_loss \leftarrow$ DAN with $\boldsymbol{H}^{\mathcal{S}}, \boldsymbol{H}^{\mathcal{T}}$ by Eq.4
            $\mathbf{F}_g^{\mathcal{S}}, \mathcal{A} \leftarrow$ calculate attention matrix and final
    representations by Eqs.5-7
            $\tilde{\mathbf{Y}}_t^{\mathcal{S}} \leftarrow Conv3D\left(\mathbf{F}^{\mathcal{S}} \oplus \mathbf{F}_g^{\mathcal{S}}\right)$
            $\mathcal{L}$++
        end if
        $t \leftarrow t + L^{\mathcal{T}} \times \mathcal{L}$
        update $\Theta$ based on Eq.8
    end while
    return $\Theta$
end while
```

$\boldsymbol{F}_g^{\mathcal{S}}$ encodes the global spatial dependency. To also encode the local spatial dependency, we add $\boldsymbol{F}^{\mathcal{S}}$ and $\boldsymbol{F}_g^{\mathcal{S}}$ to form the final feature representation $\boldsymbol{F}_{lg}^{\mathcal{S}}$ as shown in the rightmost of Fig. 1. Finally, $\boldsymbol{F}_{lg}^{\mathcal{S}}$ is fed into a linear regression model implemented by a $Conv3D$ layer for prediction.

External Features: As external context features including holidays and weather can significantly affect human mobility in urban area [1], [18], we also incorporate such features into our prediction model. As shown in the rightmost part of Fig. 1, the external context features are input into a fully connected layer for feature learning, and then they are concatenated with the model features for final prediction.

D. Overall Loss Function

The final overall loss function of ST-DAAN is as follows:

$$Loss = \frac{1}{L^{\mathcal{S}}} \sum_{t=1}^{L^{\mathcal{S}}} (\left\| \tilde{\mathbf{Y}}_t^{\mathcal{S}} - \mathbf{Y}_t^{\mathcal{S}} \right\|_F^2 + \gamma \sum_i mmd\_loss_i). \quad (8)$$

where $L^{\mathcal{S}}$ is the training sample size in the source domain, i is the layer indices of $Conv3D$, and $\gamma > 0$ is the domain discrepancy penalty parameter. The pseudo-code of the algorithm is shown in Algorithm 1.

V. Evaluation

In this section, we conduct extensive experiments over five large urban spatio-temporal datasets to evaluate the performance of the model. More specifically, we aim to answer the following questions through evaluation.

- Whether ST-DAAN can effectively transfer spatio-temporal knowledge and achieve desirable performance with scarce target domain spatio-temporal data? Whether the designed global spatial attention can better capture the spatial dependency and improve the performance?
- How the amount of available data in both target domain and source domain affect the model performance?
- Whether and to what extent ST-DAAN is sensitive to the model structure and parameters including the number of network layers and the parameter γ?

A. Experiment Setup and Datasets

We use the following five datasets for evaluations: TaxiNYC, CitiBikeNYC, BikeChicago, DiDi, and TaxiBJ datasets. We conduct the crowd flow prediction task, which is widely studied in urban spatio-temporal data prediction [1], [9], [15], [40]. Given the historical bike or taxi trips, crowd flow prediction aims to predict the outflow and inflow of the trips in each cell region in the next timestamp. The statistics of the dataset is shown in Table I, and the detailed descriptions on the datasets are as follows.

TaxiNYC dataset This data contains over 160 million taxicab trip records in New York from January 2015 to December 2015. On average, there are about 13 million trip records each month. Each taxi trip record includes fields capturing pick-up and drop-off dates/times, pick-up and drop-off locations, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts.

BikeNYC dataset This dataset contains more than 9 million bike trips in New York from January 2015 to December 2015. In total, CitiBike has established over 600 stations and 10,000 bikes in New York. Each bike trip contains the trip duration, start/end station IDs, start/end timestamps, station Lat/Long and bike ID.

BikeChicago dataset This dataset is also a bike trip data collected from the bike sharing system Divvy in Chicago. It contains more than 6 million bike trips in Chicago from January 2015 to December 2015. Divvy has 580 stations and 5,800 bikes in total. Similar to the BikeNYC dataset, each bike trip in BikeChicago also contains the start/end trip time and stations.

DiDi dataset This dataset is a taxi trip data provided by Data Center of Didi Chuxing. It contains more than 6 million taxi trips in Chengdu, China in November of 2016. Each data in this dataset contains trip ID, start/ end trip time and locations.

TaxiBJ dataset The TaxiBJ dataset is a trajectory data including the taxicab GPS data and meteorology data in Beijing. It covers the Beijing taxi trips from November of 2015 to April of 2016.

We use two types of external context features, holidays and weather. The holiday features include weekday, weekend, holidays and regular days. The weather features include temperature, wind speed, precipitation, snow, *et al.*

We choose *TaxiNYC*, *BikeChicago* as the source domain data, *BikeNYC* as the target domain data; and *BikeNYC* as

TABLE I

STATISTICS OF THE DATASETS

Dataset	TaxiNYC	BikeNYC	BikeChicago	TaxiBJ	DiDi
City	NewYork	NewYork	Chicago	Beijing	Chengdu
# of Trips	160 million	9 million	6 million	/	6 million
Latitude	[40.67, 40.77]	[40.67, 40.77]	[41.78, 41.98]	[39.85, 39.99]	[30.50, 30.80]
Longitude	[-74.02, -73.95]	[-74.02, -73.95]	[-87.71, -87.58]	[116.36, 116,50]	[103.80, 104.30]
Grid map		(16, 16)		(32, 32)	(16, 16)
Time slot		1 hour		30 minutes	1 hour
Time span		Jan 1, 2015 - Dec 31, 2015		Nov 1, 2015 - April 10, 2016	Nov 1, 2016 - Nov 30, 2016

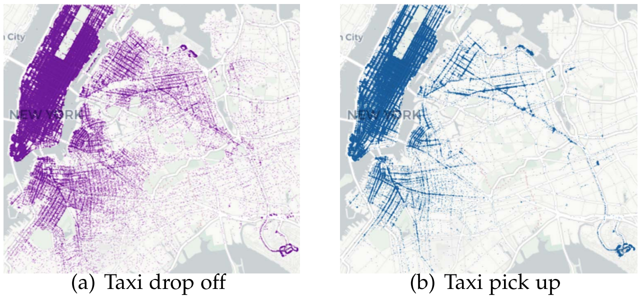

(a) Taxi drop off (b) Taxi pick up

Fig. 3. The heat maps of the TaxiNYC dataset.

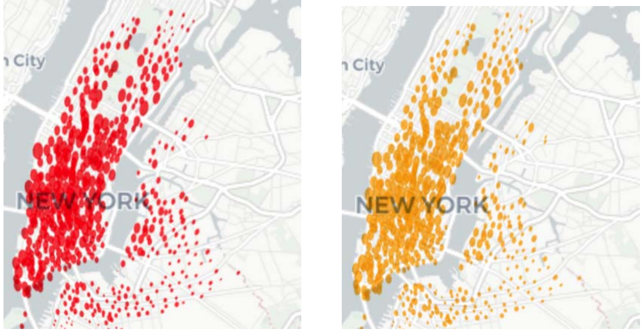

(a) CitiBike bike check out (b) CitiBike bike check in

Fig. 4. The heat maps of the CitiBikeNYC dataset.

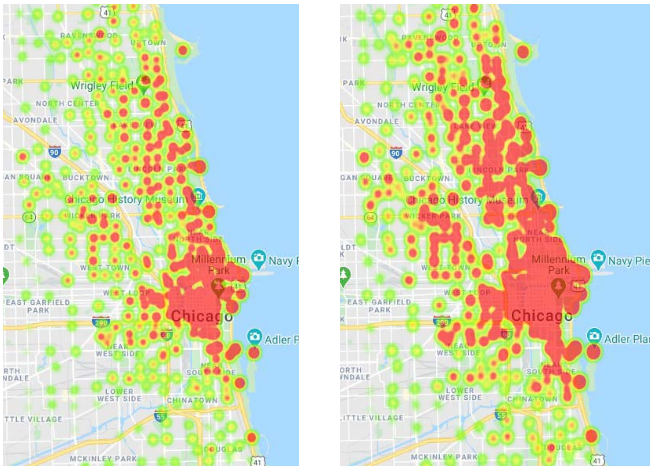

(a) Divvy bike check out (b) Divvy bike check in

Fig. 5. The heat maps of the BikeChicago dataset.

the source domain data, *TaxiNYC*, *BikeChicago* as the target domain data for evaluation, respectively. For the source domain data, we use the first 9 months data for training and validation, and the remaining 3 months data for testing. For the target domain data, we assume only 1 month data are available for training and validation, and the remaining data are used for testing. For DiDi dataset and TaxiBJ dataset, DiDi is selected as the source domain data and TaxiBJ is the target domain data. We conduct the experiment through both intra-city (*TaxiNYC→BikeNYC*) transfer and cross-city transfer (*BikeChicago→BikeNYC*, and *DiDi→TaxiBJ*). Fig. 3 shows the heat maps of the drop off and pick up locations of the taxis in New York. Fig. 4 shows the heat maps of the check out and check in locations of the CitiBike bikes in New York. Fig. 5 shows the heat maps of the BikeChicago dataset. One can see that the first two datasets are not evenly distributed in New York. There are a large number of taxi and bike trips in Manhattan, but the data are sparse in other areas of New York. As the locations of bike stations in New York are fixed and the bike trips can be only from one bike station to another, the covering areas of the bike trips is much smaller than the areas covered by the taxi trips. Similarly, the bike usage in Chicago is not evenly distributed in the city, either. Note that in some areas there are no bike or taxi flow data at all such as the ocean areas. Thus, although we still include such areas in our grid maps, we do not evaluate the prediction results for these regions as the values are always zero.

Implementation details We implement our model with Pytorch framework on GTX 1080Ti GPU. The parameters in the model are set as follows. The input data size is $6 \times 16 \times 16 \times 2$ for all the five datasets, where 6 is the previous time slot length used for prediction, 16×16 is the size of the cell regions, and 2 is the number of channels that represent inflow and outflow. The learning rate α and batch size are set to 0.001 and 24, respectively. The ConvLSTM model contains 2 layers whose structure is $6 \times 16 \times 16 \times 64$ and $6 \times 16 \times 16 \times 2$. The Conv3D module in domain adaptation part contains 2 layers whose structure is $4 \times 16 \times 16 \times 32$. The Conv3D module in global spatial attention part contains 1 layer whose structure is $1 \times 16 \times 16 \times 2$. The final Conv3D module contains 1 layer whose structure is $1 \times 16 \times 16 \times 2$. The baseline methods are implemented based on the original papers or we use the publicly available code. The parameters of baseline methods are set based on the original papers. Note that we normalize the crowd flow data in each cell region into [0,1] to facility the knowledge transfer among different types of spatio-temporal data in two domains.

The loss curve of the training process is shown in Fig. 6. We only show the loss curve with *TaxiNYC* as the source domain data and *BikeNYC* as the target domain data as an example as the curves on the other datasets present similar trends. One can see that the model converges quickly. Within around 10 epochs, the training loss first drops quickly, and then becomes stable. It shows the proposed model can quickly converge.

Evaluation metric. We adopt mean absolute error (MAE) and root mean square error (RMSE) defined as follows as the evaluation metrics.

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\left|\hat{\boldsymbol{Y}}_t - \boldsymbol{Y}_t\right|, RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left\|\hat{\boldsymbol{Y}}_t - \boldsymbol{Y}_t\right\|^2},$$

where $\hat{\boldsymbol{Y}}_t$ and $\boldsymbol{Y}_t$ are the predicted value and the corresponding ground truth in timestamp t respectively.

B. Baselines

We compare ST-DAAN with both non-transfer learning based spatio-temporal prediction models and recent transfer learning based models. For the non-transfer learning based methods, we only use the target domain data for model training. The selected non-transfer learning based prediction models for comparison are as follows.

- ***ARIMA***: Auto-Regressive Integrated Moving Average (ARIMA) [46] is a classic statistics-based method for time series prediction.
- ***ConvLSTM***: ConvLSTM [17] is a variant of LSTM which contains a convolution operation inside the LSTM cell. ConvLSTM considers both spatial and temporal dependency of the spatio-temporal data, and is widely used in many spatio-temporal prediction tasks.
- ***DCRNN***: DCRNN [19] is a deep learning framework for traffic forecasting that incorporates both spatial and temporal dependency in the traffic flow. It uses graph convolution networks to capture the spatial correlations and the the encoder-decoder architecture with scheduled sampling to capture the temporal correlations.
- ***DeepST***: [15] DeepST is a deep spatio-temporal neural network designed for urban crowd flow prediction. It uses a temporal dependent instances to capture the temporal closeness, period seasonal trend, and convolutional neural network to capture near and far spatial dependencies.
- ***ST-ResNet***: ST-ResNet [1] is a state-of-the-art deep learning framework for urban crowd flow prediction. It stacks convolutional layers and residual unites to capture the spatial dependency and both short-term and long-term temporal dependencies.

We also compare with the following transfer learning based prediction models, including fine-tuning based methods and recent state-of-the-art transfer learning models.

- ***Fine-tuning Methods*** This type of methods first use the rich spatio-temporal data of the source domain to pre-train the model, and then fine-tunes the model with the target domain data. We conduct the fine-tuning with the following methods: ConvLSTM, DeepST, DCRNN and ST-ResNet. We denote the three fine-tuned models as **ConvLSTM-FT**, **DeepST-FT**, **DCRNN-FT** and **STResNet-FT**.
- ***RegionTrans***: RegionTrans [9] transfers knowledge for a data-rich source city to a data-scarce target city. It fist calculates the similarity (e.g., *Pearson* coefficient) between each target region and source region by using some service spatio-temporal data. Such inter-city similar-regions are then used as guidance information for transfer learning and are added into a deep learning based prediction model as constraint. Note that the two stages are separated, while our ST-DAAN performs prediction and knowledge transfer in an end-to-end way.
- ***MetaST***: MetaST [10] designs a meta-learning paradigm for transferring spatio-temporal knowledge among multiple cities. MetaST focuses on transferring the common long-terms spatio-temporal patterns of different cities, but the short-term shared knowledge cannot be effectively leveraged.

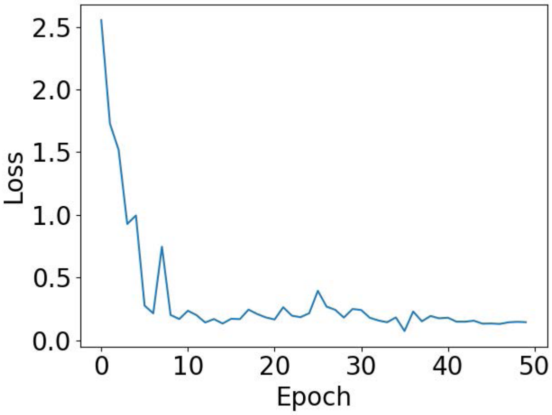


Fig. 6. The loss curve of training ST-DAAN (*TaxiNYC→BikeNYC*).

C. Experiment Results

1) Comparison With Baselines: We first compare the prediction performance of different methods in terms of MAE and RMSE. The result is shown in Table II. *TaxiNYC→BikeNYC* means that TaxiNYC is used as source domain data and BikeNYC is the target domain data; *BikeChicago→BikeNYC* means that BikeChicago is selected as the source domain while BikeNYC is the target domain. Note that only one result is shown in the table for non-transfer learning methods as the target domain data is the same in two experiments. The best results are highlighted in bold font. One can see that the proposed ST-DAAN achieves the best results in most cases with only one exception (0.0579 MAE achieved by ST-DAAN(nonAtt) over *DiDi→TaxiBJ*).

It shows that traditional statistics based method ARIMA achieves the worse performance among all the methods in both cases. It is not surprising because ARIMA only uses the time series data of each region, but ignores the spatial dependency. As a popular model for spatio-temporal data feature learning, ConvLSTM performs the best among all the non-transfer learning based methods. However, it is still inferior to the transfer learning models including both fine-tuning based methods and RegionTrans, MetaST. This verifies that transfer learning does improve the prediction performance when the source domain knowledge is properly leveraged. Although fine-tuning based methods performs better than the non-transfer learning methods, they are inferior to RegionTrans, MetaST and ST-DAAN. This

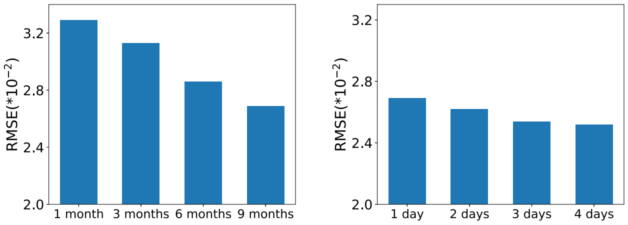
TABLE II

PERFORMANCE COMPARISON AMONG DIFFERENT METHODS (THE RAW INPUT DATA IN EACH CELL REGION IS NORMALIZED TO [0,1])

Method	*DiDi→TaxiBJ*		*TaxiNYC→BikeNYC*		*BikeChicago→BikeNYC*		*BikeNYC→TaxiNYC*		*BikeNYC→BikeChicago*	
	RMSE	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**
Target only										
ARIMA	0.2771	0.1823	0.114	0.0513	-	-	0.0712	0.0512	0.0821	0.0523
ConvLSTM	0.1279	0.0836	0.0379	0.0182	-	-	0.0523	0.0234	0.0573	0.0253
DeepST	0.1747	0.1331	0.0432	0.0189	-	-	0.0582	0.0256	0.0486	0.0184
STResNet	0.198	0.1393	0.0443	0.0198	-	-	0.0478	0.0172	0.0512	0.0201
Source+Target										
ConvLSTM-FT	0.106	0.0733	0.0378	0.0181	.0367	0.0195	0.0387	0.0192	0.0404	0.0196
DeepST-FT	0.1557	0.0778	0.0368	0.0204	0.0345	0.0178	0.0376	0.0186	0.0412	0.0206
STResNet-FT	0.1415	0.0904	0.0382	0.0202	0.0375	0.0165	0.0412	0.0212	0.0375	0.0178
RegionTrans	0.0711	0.0941	0.0298	0.0181	0.0327	0.0254	0.0301	0.0176	0.0346	0.0188
DCRNN-FT	0.1220	0.0945	0.0359	0.0177	0.0280	0.0165	0.0353	0.0186	0.0389	0.0224
MetaST	0.0795	0.0899	0.0352	0.0257	0.0344	0.0270	0.0376	0.0201	0.0412	0.0192
ST-DAAN(nonAtt)	0.0638	**0.0579**	0.0281	0.0210	0.0262	0.0174	0.0287	0.0182	0.0283	0.0176
ST-DAAN(NonExt)	0.0624	0.0756	0.0269	0.0170	0.0255	0.0122	0.0256	0.0178	0.0276	0.0168
ST-DAAN	**0.0604**	0.0729	**0.0264**	**0.0154**	**0.0210**	**0.0105**	**0.0249**	**0.0166**	**0.0274**	**0.0162**

is mainly because fine-tuning based methods directly use the learned parameters from the source domain without considering whether they are transferable or not. ST-DAAN performs the best among all the methods, and outperforms state-of-the-art methods RegionTrans and MetaST by a large margin on both datasets. For *TaxiNYC→BikeNYC*, ST-DAAN reduces RMSE by 11.4% (from 0.0298 to 0.0264) compared with the best result achieved by the baseline RegionTrans. For *BikeChicago→BikeNYC*, RMSE is reduced by around 25% (from 0.0280 to 0.0210) compared with the best result achieved by the baseline DCRNN, which is a more significant performance improvement.

Note that our model can be used in both intra-city and cross-city transfer learning. The experiment on *TaxiNYC→BikeNYC* and *BikeNYC→TaxiNYC* tests the intra-city (*Chicago→Chicago*) transfer learning performance, while the experiment on *DiDi→TaxiBJ*, *BikeChicago→BikeNYC* and *BikeNYC→BikeChicago* tests the cross-city (*Chicago↔New York*) learning performance. On both cases, ST-DAAN outperforms all the baselines, which verifies the generality of ST-DAAN. To test whether the proposed global spatial attention mechanism is useful, we compare ST-DAAN with ST-DAAN(nonAtt). ST-DAAN(nonAtt) is a variant of ST-DAAN that removes the global spatial attention mechanism. It shows that RMSE and MAE are further reduced in both experiments by using the global spatial attention (MAE is reduced from 0.021 to 0.0154 on *TaxiNYC→BikeNYC*, and from 0.0174 to 0.0105 on *BikeChicago→BikeNYC*). Thus it verifies that the proposed attention better captures the spatial correlations between the spatio-temporal data of two domains. The results over *DiDi→TaxiBJ*, *BikeNYC→TaxiNYC* and *BikeNYC→BikeChicago* show the similar result. To study whether external features are helpful to the studied problem, we also compare ST-DAAN with ST-DAAN(nonAtt). ST-DAAN(nonAtt) is a variant of ST-DAAN that removes the external features. It shows that the performance of ST-DAAN(nonAtt) is slightly inferior to ST-DAAN in most cases, which implies that external features are also useful for crowd flow prediction.

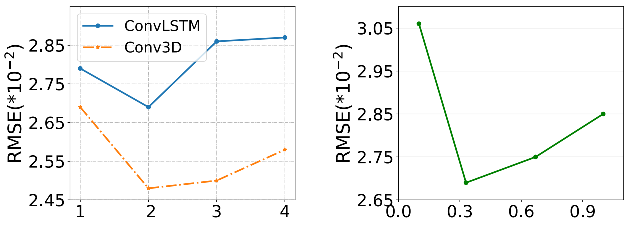


(a) # of source domain data (b) # of target domain data

Fig. 7. RMSE with variant amount of source domain data and target domain data over *TaxiNYC→BikeNYC*.

2) Effect of Data Amount: To study the effect of data amount on the model performance, we conduct experiment by sampling different amount of data in both source domain and target domain for model training. We fix the data amount of one domain, and then tune the data amount of the other domain. To test how the source domain data amount influences the prediction performance on the target domain, we first conduct the experiment with 1 month, 3 months, 6 months, and 9 months source domain data for training over *TaxiNYC→BikeNYC*. The RMSE result is shown in Fig. 7(a). One can see that the RMSE decreases from around 0.033 to around 0.027 with the increase of the training data amount in source domain. It shows that more source domain data can lead to better prediction performance on target domain as more useful knowledge can be transferred. To further test how the target domain data amount influence the model performance, we also sample 1 day, 2 days, 3 days, and 4 days target domain data, and test the performance. The RMSE shown in Fig. 7(b) also presents a decreasing trend, demonstrating more training data in target domains also leads to better performance.

3) Parameter Sensitivity Analysis: We next study how sensitive the model is on the deep neural network structure and the parameter γ. We only give the result on *TaxiNYC→BikeNYC* for an example. Fig. 8(a) shows the performance curves under different number of ConvLSTM layers and Conv3D layers. One can see that both curves first drop significantly and them slightly rise up. It shows 2 layers of ConvLSTM and Conv3D is a reasonable setting in this experiment. More layers will lead



(a) # of layers

(b) value of γ

Fig. 8. Parameter sensitivity analysis.

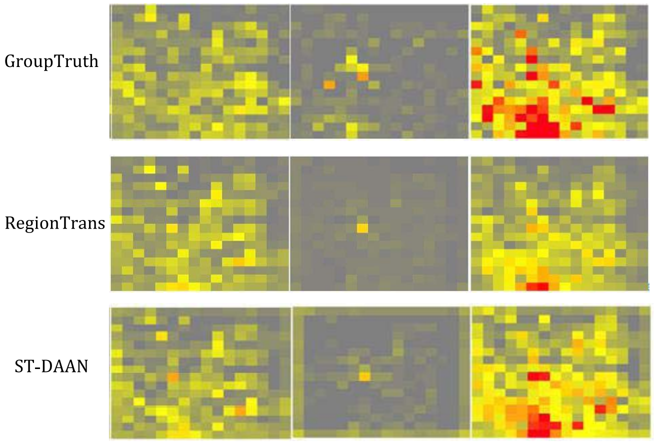


Fig. 9. The case study of the prediction results of RegionTrans and ST-DAAN in three time intervals 6:00-7:00 am (left), 22:00-23:00 pm (middle), and 15:00-16:00 pm (right).

to a more complex model with more parameters, and thus may result in over-fitting when the data is scarce. Fig. 8(b) shows the performance curve with different γ values. A larger γ means a larger penalty on the domain discrepancy, and thus more knowledge will be transferred from source domain to target domain. Fig. 8(b) shows a reasonable setting for γ is 0.3 on *TaxiNYC→BikeNYC* dataset. Too small a γ will lose some common knowledge that can be transferred, while too large a γ will let more domain-specific features be transferred. Both cases will hurt the prediction performance.

4) Case Study: To further intuitively show the effectiveness of the proposed model, we give a case study in Fig. 9. It shows the heat maps of the predicted crowd flows of TaxiBJ dataset by the RegionTrans method and our ST-DAAN method and also the ground truth. We select three time intervals on November 29, 6:00-7:00 am, 22:00-23:00 pm, and 15:00-16:00 pm, respectively. Two are rush hours and one is in deep night. One can see that the crowd flows in the three hours are quite different. There are a large number of people taking taxis in rush hours, while the requirement of taxis in the deep night drops significantly. Compared with RegionTrans, the prediction of ST-DAAN is more close to the ground truth, especially in the time interval 15:00-16:00 pm. The case study also shows that our proposed ST-DAAN performs better than RegionTrans method.

VI. Conclusion and Future Work

This paper proposes a deep attentive adaptation network based transfer learning framework named ST-DAAN for urban crowd flow data prediction. The novelty of the models lies in the usage of deep adaptation network to match the domain distributions in the embedded latent space. An end-to-end solution is proposed to effectively learning transferable latent features for cross-domain urban spatiao-temporal data prediction to address the data scarcity issue. To effectively capture the complex cross-domain spatial dependency, a global spatial attention mechanism is also designed. We evaluate the proposed model on five real large datasets. The results demonstrate the superior performance of the proposed model on both intra- and cross-city spatio-temproal data transfer learning. As the proposed ST-DAAN model is general, it can be easily extended and generalized to other spatio-temporal prediction tasks. In the future, it would be interesting to further apply ST-DAAN to the tasks of traffic flow prediction and POI recommendation.

References

[1] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[2] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[3] M. M. Aburas, Y. M. Ho, M. F. Ramli, and Z. H. Ash'aari, "The simulation and prediction of spatio-temporal urban growth trends using cellular automata models: A review," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 52, pp. 380–389, Oct. 2016.

[4] X. Zhou, Y. Shen, Y. Zhu, and L. Huang, "Predicting multi-step citywide passenger demands using attention-based neural networks," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2018, pp. 736–744.

[5] J. Ye, L. Sun, B. Du, Y. Fu, X. Tong, and H. Xiong, "Co-prediction of multiple transportation demands based on deep spatio-temporal neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 305–313.

[6] B. Du *et al.*, "Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 972–985, Mar. 2020.

[7] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," 2018, *arXiv:1803.01254*. [Online]. Available: http://arxiv.org/abs/1803.01254

[8] X. Cheng, R. Zhang, J. Zhou, and W. Xu, "DeepTransport: Learning spatial-temporal dependency for traffic condition forecasting," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2018, pp. 1–8.

[9] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang, "Cross-city transfer learning for deep spatio-temporal prediction," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1893–1899.

[10] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, "Learning from multiple cities: A meta-learning approach for spatial-temporal prediction," in *Proc. World Wide Web Conf. (WWW)*, 2019, pp. 2181–2191.

[11] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[12] S. Shekhar and B. M. Williams, "Adaptive seasonal time series models for forecasting short-term traffic flow," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2024, no. 1, pp. 116–125, Jan. 2007.

[13] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, Jun. 2013.

[14] J. Zheng and L. M. Ni, "Time-dependent trajectory regression on road networks via multi-task learning," in *Proc. 27th AAAI Conf. Artif. Intell.*, 2013, pp. 1048–1055.

[15] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Oct. 2016, pp. 2588–2595.

[16] S. Wang, J. Cao, H. Chen, H. Peng, and Z. Huang, "Seqst-GAN: Seq2seq generative adversarial nets for multi-step urban crowd flow prediction," *ACM Trans. Spatial Algorithms Syst.*, vol. 6, no. 4, pp. 1–22, 2020.

[17] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1–9.

[18] S. Wang, H. Miao, H. Chen, and Z. Huang, "Multi-task adversarial spatial-temporal networks for crowd flow prediction," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1555–1564.

[19] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. ICLR Conf.*, 2018, pp. 1–16.

[20] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 468–478, Mar. 2020.

[21] Y. Liang *et al.*, "UrbanFM: Inferring fine-grained urban flows," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 3132–3142.

[22] Z. Pan, Z. Wang, W. Wang, Y. Yu, J. Zhang, and Y. Zheng, "Matrix factorization for spatio-temporal neural networks with applications to urban flow prediction," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2683–2691.

[23] P. Xie, T. Li, J. Liu, S. Du, X. Yang, and J. Zhang, "Urban flow prediction from spatiotemporal data using machine learning: A survey," *Inf. Fusion*, vol. 59, pp. 1–12, Jul. 2020.

[24] S. Wang, J. Cao, and P. Yu, "Deep learning for spatio-temporal data mining: A survey," *IEEE Trans. Knowl. Data Eng.*, early access, Sep. 22, 2020, doi: 10.1109/TKDE.2020.3025580.

[25] L. Wang, B. Guo, and Q. Yang, "Smart city development with urban transfer learning," *Computer*, vol. 51, no. 12, pp. 32–41, Dec. 2018.

[26] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[27] F. Zhuang *et al.*, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.

[28] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[29] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised representation learning with double encoding-layer autoencoder for transfer learning," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 2, pp. 1–17, Jan. 2018.

[30] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.

[31] Q. Do, W. Liu, J. Fan, and D. Tao, "Unveiling hidden implicit similarities for cross-domain recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 1, pp. 302–315, Jan. 2020.

[32] B. Guo, J. Li, V. W. Zheng, Z. Wang, and Z. Yu, "CityTransfer: Transferring inter- and intra-city knowledge for chain store site recommendation based on multi-source urban data," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 1–23, Jan. 2018.

[33] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.

[34] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 994–1003.

[35] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster R-CNN for object detection in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 994–1003.

[36] E. Hosseini-Asl, R. Keynton, and A. El-Baz, "Alzheimer's disease diagnostics by adaptation of 3D convolutional network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 126–130.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] Z. Yuan, X. Zhou, and T. Yang, "Hetero-ConvLSTM: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 984–992.

[40] L. Liu, R. Zhang, J. Peng, G. Li, B. Du, and L. Lin, "Attentive crowd flow machines," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 1553–1561.

[41] S. Kim, S. Hong, M. Joh, and S.-K. Song, "DeepRain: ConvLSTM network for precipitation prediction using multichannel radar data," 2017, *arXiv:1711.02316*. [Online]. Available: http://arxiv.org/abs/1711.02316

[42] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, Mar. 2012.

[43] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[44] S. Wang, H. Chen, J. Cao, J. Zhang, and P. S. Yu, "Locally balanced inductive matrix completion for demand-supply inference in stationless bike-sharing systems," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 12, pp. 2374–2388, Dec. 2020.

[45] H. Yao *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 2588–2595.

[46] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. OTexts, 2014. [Online]. Available: https://otexts.com

Senzhang Wang (Member, IEEE) received the B.Sc. degree from Southeast University, Nanjing, China, in 2009, and the Ph.D. degree from Beihang University, Beijing, China, in 2016. He is currently a Professor with the School of Computer Science and Engineering, Central South University. His main research focus is on spatiotemporal data mining, graph data mining, and urban computing. He has published more than 90 referred conference and journal papers.

Hao Miao received the B.S. degree in computer science and technology from Nanjing Tech University, Nanjing, China, in 2018. He is currently pursuing the master's degree with the Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. From September 2019 to November 2019, he was a Visiting Student with Hong Kong Polytechnic University, Hong Kong. His research interests include spatiotemporal data mining, deep learning, and transfer learning.

Jiyue Li received the B.S. degree in software engineering from Guangzhou University, Guangzhou, China, in 2020. She is currently pursuing the master's degree with the Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics. In July 2019, she was a Visiting Student with the University of Washington, Seattle, USA. Her research interests include spatiotemporal data mining and deep learning.

Jiannong Cao (Fellow, IEEE) received the B.Sc. degree in computer science from Nanjing University, China, in 1982, and the M.Sc. and Ph.D. degrees in computer science from Washington State University, USA, in 1986 and 1990 respectively. He is currently a Chair Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His research interests include parallel and distributed computing, wire-less networks and mobile computing, big data and cloud computing, pervasive computing, and fault tolerant computing. He has coauthored five books in mobile computing and wireless sensor networks, co-edited none books, and published over 500 papers in major international journals and conference proceedings. He is a Distinguished Member of ACM and a Senior Member of China Computer Federation (CCF).