Cross-City Transfer Learning for Deep Spatio-Temporal Prediction

Leye Wang[1,2], **Xu Geng**[2], **Xiaojuan Ma**[2], **Feng Liu**[3] and **Qiang Yang**[2,4]
[1]Key Lab of High Confidence Software Technologies, Peking University, Ministry of Education, China
[2]Hong Kong University of Science and Technology, Hong Kong, China
[3]SAIC Motor, Shanghai, China
[4]WeBank, Shenzhen, China
leyewang@pku.edu.cn, xgeng@connect.ust.hk, mxj@cse.ust.hk, liufeng@saicmotor.com, qyang@cse.ust.hk

Abstract

Spatio-temporal prediction is a key type of tasks in urban computing, e.g., traffic flow and air quality. Adequate data is usually a prerequisite, especially when deep learning is adopted. However, the development levels of different cities are unbalanced, and still many cities suffer from data scarcity. To address the problem, we propose a novel cross-city transfer learning method for deep spatio-temporal prediction tasks, called *RegionTrans*. *RegionTrans* aims to effectively transfer knowledge from a data-rich source city to a data-scarce target city. More specifically, we first learn an inter-city region matching function to match each target city region to a similar source city region. A neural network is designed to effectively extract region-level representation for spatio-temporal prediction. Finally, an optimization algorithm is proposed to transfer learned features from the source city to the target city with the region matching function. Using crowd flow prediction as a demonstration experiment, we verify the effectiveness of *RegionTrans*.

1 Introduction

Spatio-temporal prediction covers a broad scope of applications in urban computing [Zheng *et al.*, 2014], such as traffic and air quality prediction. Recently, with the development of big data techniques, deep learning becomes popular in spatio-temporal prediction, e.g. crowd flow, taxi demand, precipitation predictions, and achieves state-of-the-art performance [Ke *et al.*, 2017; Shi *et al.*, 2015; Zhang *et al.*, 2017; Zhang *et al.*, 2016]. However, the city development levels are quite unbalanced, so that many cities cannot benefit from such achievements due to data scarcity. Hence, how to help data-scarce cities also obtain benefits from the recent technique breakthroughs like deep leaning, becomes an important research issue, while it is still under-investigated up to date.

To tackle this problem, in this paper, we propose a new cross-city transfer learning method for deep spatio-temporal prediction tasks, called *RegionTrans*. The objective of *RegionTrans* is to predict a certain type of service data (e.g., crowd flow) in a data-scarce city (*target city*) by transferring knowledge learned from a data-rich city (*source city*). The principal idea of *RegionTrans* is to find *inter-city region pairs* that share similar patterns and then use such region pairs as proxies to efficiently transfer knowledge from the source city to the target.

In literature, existing deep learning approaches are often designed to predict citywide phenomenon as a whole [Zhang *et al.*, 2017; Zhang *et al.*, 2016], and thus it is hard to enable region-level knowledge transfer. To this end, rather than adopting the existing deep neural networks for citywide spatio-temporal prediction, e.g. ST-ResNet [Zhang *et al.*, 2017], we propose a novel deep transfer learning method. First, we design a region matching function to link each target city region to a similar source region based on the short period of service data or correlated auxiliary data if applicable. Then, in our proposed network structure, to catch the spatio-temporal patterns hidden in the service data, ConvLSTM layers [Shi *et al.*, 2015] are firstly stacked. Afterward, to encode *region representation*, we newly add a Conv2D layer with 1×1 filter, which is the key and fundamental component of our network to make region-level transfer feasible. Finally, the discrepancy between *region representations* of the inter-city similar regions is minimized during the network parameter learning for the target city, so as to enable region-level cross-city knowledge transfer. With crowd flow prediction as a showcase [Zhang *et al.*, 2017], we verify the feasibility and effectiveness of *RegionTrans*.

Briefly, this paper has the following contributions.

(i) To the best of our knowledge, this is the first work to facilitate deep spatio-temporal prediction in a data-scarce target city by transferring knowledge from a data-rich source city.

(ii) We propose a novel deep transfer learning method *RegionTrans* for spatio-temporal prediction tasks by region-level cross-city transfer. *RegionTrans* first computes inter-city region similarities, and then stacks ConvLSTM and Conv2D (1×1 filter) layers to extract region-level representations reflecting spatio-temporal patterns. Finally, the discrepancy of the representations of inter-city similar regions is minimized so as to facilitate region-level cross-city knowledge transfer.

(iii) With crowd flow prediction as a showcase, our experiment shows that *RegionTrans* can reduce up to 10.7% predic-

This research is partially supported by Hong Kong ITF project no. ITS/391/15FX, and the collaboration project at HKUST-DiDi Joint Lab.

tion error compared to fine-tuned state-of-the-art deep spatio-temporal prediction methods.

2 Related Work

Spatio-Temporal Prediction is a fundamental problem in urban computing [Zheng *et al.*, 2014]. Recently, deep learning is adopted in spatio-temporal prediction tasks and becomes the state-of-the-art solution when there exists a rich history of data. Various deep models have been used, e.g., CNN [Zhang *et al.*, 2016], ResNet [Zhang *et al.*, 2017], and ConvLSTM [Ke *et al.*, 2017; Shi *et al.*, 2015; Yao *et al.*, 2018]. Compared to these works, the difference of our work lies in both objective and method. We aim to apply deep learning to a target city with a short period of service data, and thus propose *RegionTrans* to effectively transfer knowledge from a data-rich source city to the target city.

Transfer Learning addresses the machine learning problem when labeled training data is scarce [Pan and Yang, 2010]. In urban computing, data scarcity problem often exists when the targeted service or infrastructure is new. There are generally two strategies to deal with urban data scarcity. The first is using auxiliary data of the target city to help build the targeted application. Examples include using temperature to infer humidity and vice versa [Wang *et al.*, 2017], and leveraging the taxi GPS traces to detect ridesharing cars [Wang *et al.*, 2019]. The second is to find a source city with adequate data to transfer knowledge. Guo et al. design a cross-city transfer learning framework with collaborative filtering and AutoEncoder to conduct chain store site recommendation [Guo *et al.*, 2018]. As our problem is prediction rather than recommendation, the method in [Guo *et al.*, 2018] cannot be applied. Another relevant work is [Wei *et al.*, 2016], which proposes a cross-city transfer learning algorithm FLORAL to predict air quality category. There are two difficulties to apply FLORAL to our task: (1) many spatio-temporal prediction tasks are regression but FLORAL is designed for classification; (2) FLORAL is not designed for deep learning. As far as we know, *RegionTrans* is the first cross-city transfer learning framework for deep spatio-temporal prediction.

3 Problem Formulation

Definition 1. Region. [Zhang *et al.*, 2016] A city $\mathcal{D}$ is partitioned into $W_{\mathcal{D}} \times H_{\mathcal{D}}$ equal-size grids (e.g., $1km \times 1km$). Each grid is called a *region*, denoted as r. We use $r_{[i,j]}$ to represent a city region whose coordinate is $[i, j]$. The whole set of regions in a city $\mathcal{D}$ is denoted as $\mathbb{C}_{\mathcal{D}}$.

Definition 2. Urban Image Time Series. We denote the *set of data time-stamps* of a city $\mathcal{D}$ as:

$$\mathbb{T}_{\mathcal{D}} = [t_c - T_{\mathcal{D}} + 1, t_c] \tag{1}$$

where $T_{\mathcal{D}}$ is the number of time-stamps and t_c is the current/last time-stamp. For brevity, we consider equal-length time-stamp (e.g., one-hour) as in the previous research [Zhang *et al.*, 2017; Zhang *et al.*, 2016]. For a specific time-stamp t, we have an *urban image* $\mathcal{I}_{t,\mathcal{D}}$ with $W_{\mathcal{D}} \times H_{\mathcal{D}}$ pixels where each pixel represents certain data of a corresponding region (Def. 1),

$$\mathcal{I}_{t,\mathcal{D}} = \{i_{r,t} | r \in \mathbb{C}_{\mathcal{D}}\} \in \mathbb{R}^{W_{\mathcal{D}} \times H_{\mathcal{D}}} \tag{2}$$

Then, we define an *urban image time series* $\mathbb{I}_{\mathcal{D}}$ as follows:

$$\mathbb{I}_{\mathcal{D}} = \{\mathcal{I}_{t,\mathcal{D}} | t \in \mathbb{T}_{\mathcal{D}}\} \in \mathbb{R}^{T_{\mathcal{D}} \times W_{\mathcal{D}} \times H_{\mathcal{D}}} \tag{3}$$

In reality, a variety of urban data can be modeled as the above urban image time series, such as crowd flow, weather condition, air quality, etc.

Definition 3. Service Spatio-temporal Data. Service data is the targeted type of data to predict. We define the service spatio-temporal data as the urban image time series $\mathbb{S}^{\mathcal{D}}$ storing the service data :

$$\begin{aligned} \mathbb{S}_{\mathcal{D}} &= \{\mathcal{S}_{t,\mathcal{D}} | t \in \mathbb{T}_{\mathcal{D}}\} \\ &= \{s_{r,t} | r \in \mathbb{C}_{\mathcal{D}}, t \in \mathbb{T}_{\mathcal{D}}\} \in \mathbb{R}^{T_{\mathcal{D}} \times W_{\mathcal{D}} \times H_{\mathcal{D}}} \end{aligned} \tag{4}$$

where $s_{r,t}$ is the service data of region r at time-stamp t.

In this paper, the target city $\mathcal{D}$ suffers from the service data scarcity, while the source city $\mathcal{D}'$ has rich service data, i.e., $|\mathbb{T}_{\mathcal{D}}| \ll |\mathbb{T}_{\mathcal{D}'}|$. With this in mind, we formulate the problem.

Problem of Cross-City Spatio-temporal Prediction. Given the little service data in target city $\mathcal{D}$ and rich service data in source city $\mathcal{D}'$, we aim to learn a function f to predict the citywide service data in the target city $\mathcal{D}$ at the next time-stamp $t_c + 1$:

$$\min_{f} \quad error(\widetilde{\mathcal{S}}_{t_c+1,\mathcal{D}}, \mathcal{S}_{t_c+1,\mathcal{D}}) \tag{5}$$

$$\text{where} \quad \widetilde{\mathcal{S}}_{t_c+1,\mathcal{D}} = f(\mathbb{S}_{\mathcal{D}}, \mathbb{S}_{\mathcal{D}'}), \quad |\mathbb{T}_{\mathcal{D}}| \ll |\mathbb{T}_{\mathcal{D}'}| \tag{6}$$

error metric may be mean absolute error, root mean squared error, etc., according to the real application requirement.

Example. Crowd Flow Prediction. We use crowd flow prediction [Zhang *et al.*, 2017; Zhang *et al.*, 2016] as an example to illustrate the above problem concretely. The service data $\mathbb{S}_{\mathcal{D}}$ is thus crowd inflow or outflow. The source city crowd flow records may last for several years ($\mathbb{T}_{\mathcal{D}'}$), but the target city may have only a few days ($\mathbb{T}_{\mathcal{D}}$) as the service is just started. It is worth noting that external context factors, such as weather and workday/weekend, are also important in crowd flow prediction [Zhang *et al.*, 2017]. Later we will show that our proposed method is easy to add the external features extracted from context factors.

4 RegionTrans

4.1 Overview

Fig. 1 gives an overview of the RegionTrans framework. In brief, *RegionTrans* consists of three novel components.

1. **Inter-city similar-region matching**. We propose two ways to match region pairs between source and target cities. The first one is directly using the short period of service data in the target city to find the similar region in the source city. However, sometimes directly calculating the similarity between a source region and a target region using the short service data may not yield robust results. Suppose that the service data is crowd flow, the target city only has one day of crowd flow history and it happened to be a rainy day, but it rarely rains in the source city. Apparently, using such crowd flow data of the source and target city to compute inter-city region

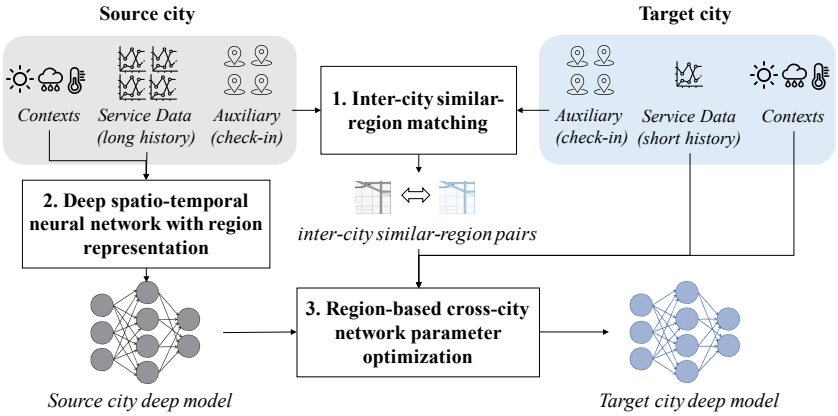

Figure 1: Overview of RegionTrans.

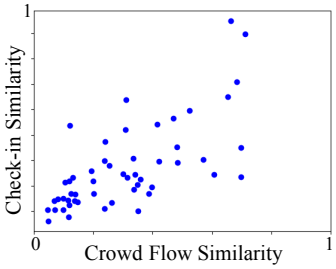

Figure 2: Check-in/crowd flow similarities.

similarity is inadequate. Hence, we propose a second method to rely on auxiliary data (if applicable) to find more robust inter-city similar region pairs. For example, widely-existing social check-ins may be the auxiliary data to indicate the crowd flow dynamics and thus help region matching.

2. **Deep spatio-temporal neural network with region representations**. Existing literature has proposed a few deep models for predicting citywide crowd flow [Zhang *et al.*, 2016; Zhang *et al.*, 2017]. However, these models usually predict citywide crowd flow as a whole, and thus are hard to incorporate region similarity information for transfer learning. Therefore, we propose a new deep spatio-temporal neural network structure, in which a 'region-representation' layer is dedicatedly designed to preserve region-level features. Based on this neural network, we then learn a source city spatio-temporal prediction model from its long historical record of service data and corresponding contexts (e.g., weather). This source city model will be later used in transfer learning for building the target city model.
3. **Region-based cross-city network parameter optimization**. Based on the deep model with region representations, we propose a cross-city network parameter optimization algorithm to learn the crowd flow prediction model for the target city, considering the source city model, the inter-city similar-region pairs, and the short period of crowd flow data of the target city.

4.2 Inter-city Similar-region Matching

The first step of *RegionTrans* is to find a matching function $\mathcal{M} : \mathbb{C}_{\mathcal{D}} \rightarrow \mathbb{C}_{\mathcal{D}'}$ to map each region of the target city $\mathcal{D}$ to a certain region of the source city $\mathcal{D}'$. The objective is to find the source region having the similar spatio-temporal pattern with the target region. We propose two strategies to find $\mathcal{M}$.

Matching with a Short Period of Service Data

While the target city has only a little service data, this could still provide hints to build $\mathcal{M}$. We focus on the time span when both source and target cities have service data (i.e., $\mathbb{T}_{\mathcal{D}}$), then calculate the correlations (e.g., *Pearson* coefficient) between each target region and source region with the corresponding service data. Finally, for each target region, we choose the source region with the largest correlation value. Formally,

$$\mathcal{M}(r) = r^*, \qquad r \in \mathbb{C}_{\mathcal{D}}, r^* \in \mathbb{C}_{\mathcal{D}'}$$
$$\rho_{r,r^*} \geq \rho_{r,r'}, \qquad \forall r' \in \mathbb{C}_{\mathcal{D}'}$$
$$\rho_{r,r^*} = corr(\{s_{r,t}\}, \{s_{r^*,t}\}), \quad r \in \mathbb{C}_{\mathcal{D}}, r^* \in \mathbb{C}_{\mathcal{D}'}, t \in \mathbb{T}_{\mathcal{D}}$$

Matching with a Long Period of Auxiliary Data

As there is little service data in the target city, the above service-data-based correlation similarity between a source region and a target region may not be very reliable. In reality, sometimes we can find another openly-accessible auxiliary data that correlates with the service data, which may help calculate the inter-city region similarity more robustly. For example, to predict crowd flow, social media check-ins can be a useful proxy according to literature [Yang *et al.*, 2016]. That is, instead of the short period of crowd flow data, we use the long period of openly available check-in data to build the correlation between two regions.

$$\rho_{r,r^*} = corr(\{a_{r,t}\}, \{a_{r^*,t}\}), \quad r \in \mathbb{C}_{\mathcal{D}}, r^* \in \mathbb{C}_{\mathcal{D}'}, t \in \mathbb{T}_{\mathcal{A}}$$

where a is the auxiliary data (e.g., check-in number) lasting for a long period $\mathbb{T}_{\mathcal{A}}(|\mathbb{T}_{\mathcal{A}}| \gg |\mathbb{T}_{\mathcal{D}}|)$.

Example: Check-in as Auxiliary for Crowd Flow

For a certain region $r \in \mathbb{C}$ in a city, we can model the check-in representation according to its hourly check-in counts in workday and weekend/holiday as follows:

$$\boldsymbol{ch_r} = \langle ch_0, ch_1, \cdots, ch_{23}, ch'_0, ch'_1, \cdots, ch'_{23} \rangle, r \in \mathbb{C}$$

where ch_i is the average check-in counts in r at i^{th} hour in workday of the whole check-in historical record; ch'_i is the hourly average check-in counts in weekend/holiday.

To verify whether the similarity between check-in representations can actually reflect the similarity of crowd flow dynamics, we conduct an analysis with bikesharing data in Washington D.C. and Chicago during 2015-2016. Here, we measure the crowd flow similarity between two regions by

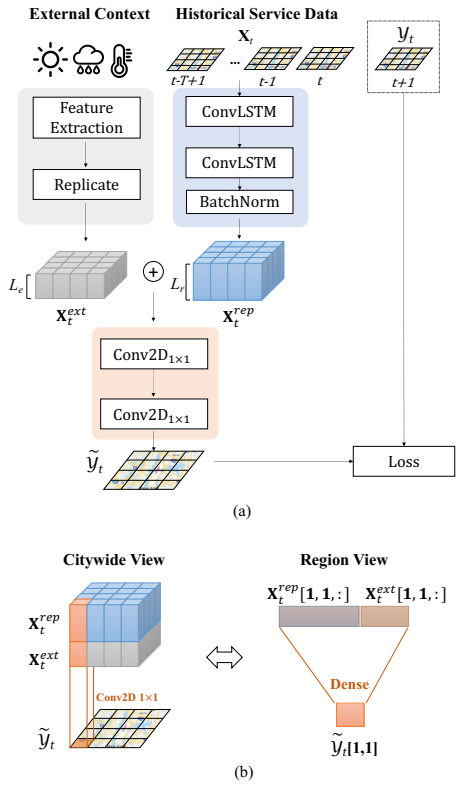

Figure 3: Proposed network structure.

first counting the hourly bike inflow/outflow counts and then use the Pearson correlation coefficient. Given a randomly selected Chicago region r^*, Fig. 2 plots both the check-in similarity (y axis) and crowd flow similarity (x axis) of every D.C. region and the selected Chicago region r^*. As expected, the D.C. regions with higher check-in similarities tend to hold higher crowd flow similarities.

4.3 Deep Spatio-Temporal Neural Network with Region Representations

Existing deep spatio-temporal models often take the whole city data for end-to-end prediction, e.g., ST-ResNet [Zhang *et al.*, 2017], which cannot be used for region-level transfer. Therefore, we design a network for spatio-temporal prediction with region representations, as shown in Fig. 3 (a).

First we illustrate the input and output of the network[1]:

$k \in \mathbb{N}^+$: the length of the input time series

$\mathbf{X}_t = \{\mathcal{S}_{t'} | t' \in [t-k+1, t]\} \in \mathbb{R}^{k \times W \times H}$: input

$\mathcal{Y}_t = \mathcal{S}_{t+1} \in \mathbb{R}^{W \times H}$: ground-truth result at time $t+1$

$f_\theta : \mathbb{R}^{k \times W \times H} \to \mathbb{R}^{W \times H}$: neural network with parameter θ

$\widetilde{\mathcal{Y}}_t = f_\theta(\mathbf{X}_t) \in \mathbb{R}^{W \times H}$: prediction result at time $t+1$

Our network objective is to minimize the squared error between predicted $\widetilde{\mathcal{Y}}_t$ and real $\mathcal{Y}_t$:

$$\min_\theta \sum_{t \in \mathbb{T}} ||\widetilde{\mathcal{Y}}_t - \mathcal{Y}_t||_F^2 \tag{7}$$

Then, our spatio-temporal network can be formulated as:

$$\textit{ConvLSTM: } f_{\theta_1} : \mathbb{R}^{k \times W \times H} \to \mathbb{R}^{W \times H \times L_r} \tag{8}$$

$$\textit{Region representation: } \mathbf{X}_t^{rep} = f_{\theta_1}(\mathbf{X}_t) \tag{9}$$

$$\textit{Merge: } f_m : (\mathbb{R}^{W \times H \times L_r}, \mathbb{R}^{W \times H \times L_e}) \to \mathbb{R}^{W \times H \times (L_r + L_e)} \tag{10}$$

$$\textit{Conv2D}_{1\times1}\textit{: } f_{\theta_2} : \mathbb{R}^{W \times H \times (L_r + L_e)} \to \mathbb{R}^{W \times H} \tag{11}$$

$$\textit{Prediction : } \widetilde{\mathcal{Y}}_t = f_{\theta_2}(f_m(\mathbf{X}_t^{rep}, \mathbf{X}_t^{ext})) \tag{12}$$

$$= f_{\theta_2}(f_m(f_{\theta_1}(\mathbf{X}_t), \mathbf{X}_t^{ext})) \tag{13}$$

ConvLSTM layers are the basic components for our network to learn spatio-temporal patterns [Shi *et al.*, 2015]. We first use a set of stacked ConvLSTM layers to construct region-level hidden representation $\mathbf{X}_t^{rep} \in \mathbb{R}^{W \times H \times L_r}$ (we will elaborate why this can be seen as region-level representation soon). After getting $\mathbf{X}_t^{rep}$, we incorporate the external context factors into the network structure. External context factors are defined as $\mathbf{X}_t^{ext} \in \mathbb{R}^{W \times H \times L_e}$, which is a feature vector of length L_e on each region (e.g., weather, temperature, weekday/holiday one-hot encoding [Zhang *et al.*, 2017]). By concatenating $\mathbf{X}_t^{rep}$ and $\mathbf{X}_t^{ext}$ to form a representation $\in \mathbb{R}^{W \times H \times (L_r + L_e)}$, we employ several convolution 2D layers with 1×1 filters ($\text{Conv2D}_{1\times1}$ [Lin *et al.*, 2014]) to predict the next-time-stamp service data $\widetilde{\mathcal{Y}}_t \in \mathbb{R}^{W \times H}$.

As visualized by Fig. 3 (b), $\text{Conv2D}_{1\times1}$ will produce spatio-invariant results, which means hidden vector $\mathbf{X}_t^{rep}[w, h, :]$ and prediction $\widetilde{\mathcal{Y}}_t[w, h]$ represent the spatio-temporal representation and prediction result of region $r_{[w,h]}$, respectively. Compared with existing end-to-end citywide deep spatio-temporal prediction models [Zhang *et al.*, 2017; Zhang *et al.*, 2016] without such region-level hidden representations, our network design has the following advantages for transfer learning:

1. *Fine-grained region-level transfer*. With existing methods which consider the data of a city as a whole for prediction, we can only transfer the knowledge from the whole source city to the target (e.g., through fine-tuning). If two cities are not similar in general, the transfer performance may be poor. As our network incorporates region representation, we can make fine-grained knowledge transfer based on region similarity (the detailed algorithm in the next sub-section). As long as we can find similar region pairs between cities, the effective transfer may be conducted.
2. *Transfer between cities with different sizes*. Since our neural network structure can be seen from region view (Fig. 3 (b)), even if two cities have different sizes (i.e. W, H), it is possible to train a model on a source city and then transfer the learned network parameters to the target city at the region level. However, with end-to-end network structures [Zhang *et al.*, 2017; Zhang *et al.*, 2016], if we want to transfer a learned model from the

[1]For clarity, we omit the subscript $\mathcal{D}$ in notations as all the notations mentioned in this section is in city $\mathcal{D}$.

source city to the target by fine-tuning, the two cities must be the same size.

4.4 Region-based Cross-city Network Parameter Optimization

With the proposed network structure, we train a deep model in the source city $\mathcal{D}'$ with its rich spatio-temporal service data. We denote $\theta_{\mathcal{D}'}$ as the network parameters learned from the source city. Then, with $\theta_{\mathcal{D}'}$ as the pre-trained network parameters, we propose a region-based cross-city optimization algorithm to refine the network parameters on the target city $\mathcal{D}$, considering a short period $\mathbb{T}_{\mathcal{D}}$ of the service data in the target city $\mathcal{D}$ and the inter-city region matching function $\mathcal{M}$.

When refining the network parameter for the target city $\mathcal{D}$, the first objective is to minimize prediction error on $\mathcal{D}$:

$$\min_{\theta_{\mathcal{D}}} \sum_{t \in \mathbb{T}_{\mathcal{D}}} ||\widetilde{\mathcal{Y}}_t - \mathcal{Y}_t||_F^2$$

Given the matching function $\mathcal{M}$, the second objective is to minimize representation divergence between matched region pairs. More specifically, for each time-stamp $t \in \mathbb{T}_{\mathcal{D}}$, we try to minimize the squared error between the network hidden representations of the target region and its matched source region. Formally, the second objective is as follows:

$$\min_{\theta_{\mathcal{D}}} \sum_{r \in \mathbb{C}_{\mathcal{D}}} \sum_{t \in \mathbb{T}_{\mathcal{D}}} \rho_{r,r^*} \cdot ||\mathbf{x}_{r,t}^{rep} - \mathbf{x}_{r^*,t}^{rep}||^2, \text{ where } r^* = \mathcal{M}(r)$$

where $\mathbf{x}_{r,t}^{rep}$ is the hidden representation of the target region r when the last input time-stamp is t; $\mathbf{x}_{r^*,t}^{rep}$ is the representation of the matched source region r^*; ρ_{r,r^*} is the correlation value calculated between region pairs (Sec. 4.2), so that more similar pair will be assigned with larger weights in the optimization. Then, combining the two objectives leads to the following optimization process:

$$\begin{aligned} \min_{\theta_{\mathcal{D}}} \quad & (1-w) \sum_{t \in \mathbb{T}_{\mathcal{D}}} ||\widetilde{\mathcal{Y}}_t - \mathcal{Y}_t||_F^2 \\ & + w \sum_{r \in \mathbb{C}_{\mathcal{D}}} \sum_{t \in \mathbb{T}_{\mathcal{D}}} \rho_{r,r^*} \cdot ||\mathbf{x}_{r,t}^{rep} - \mathbf{x}_{r^*,t}^{rep}||^2 \end{aligned} \tag{14}$$

where w is the weight to trade off between minimizing the representation discrepancy or minimizing the prediction error. Then, we can use state-of-the-art network parameter learning algorithms, such as *SGD* and *ADAM*, to obtain the network parameter $\theta_{\mathcal{D}}$ for the target city $\mathcal{D}$ according to Eq. 14 (the network parameter $\theta_{\mathcal{D}'}$ learned in the source city $\mathcal{D}'$ is used as the initialization values). The detailed pseudo-code of the optimization process is summarized in Alg. 1.

Algorithm 1 Region-based cross-city network parameter optimization

Input:
$\theta_{\mathcal{D}'}$: Pre-trained network parameters on source city with a long period of service data
$TR_{\mathcal{D}}$: target city training data
$TR_{\mathcal{D}'}$: source city training data
$\mathcal{M}$: inter-city similar-region matching function
Output:
$\theta_{\mathcal{D}}$: network parameters for the target city

Initialize network parameters: $\theta \leftarrow \theta_{\mathcal{D}'}$
epoch $\leftarrow 0$
while epoch $\leq$ MAX_EPOCH **do**
  for $t \in \mathbb{T}_{\mathcal{D}}$ **do**
    Get $\{\mathbf{X}_t, \mathcal{Y}_t\} \in TR_{\mathcal{D}}$
    Get corresponding $\{\mathbf{X}'_t, \mathcal{Y}'_t\} \in TR_{\mathcal{D}'}$
    for $r \in \mathbb{C}_{\mathcal{D}}$ **do**
      $r^* \leftarrow \mathcal{M}(r)$ (note that $r^* \in \mathbb{C}_{\mathcal{D}'}$)
      $\mathbf{x}_{r^*,t}^{rep} \leftarrow$ region representation with input $\mathbf{X}'_t$ for source r^*
      $\mathbf{x}_{r,t}^{rep} \leftarrow$ region representation with input $\mathbf{X}_t$ for target r
    end for
  end for

$$\begin{aligned} \theta \leftarrow \arg\min_{\theta} \quad & (1-w) \sum_{t \in \mathbb{T}^{\mathcal{D}}} ||\widetilde{\mathcal{Y}}_t - \mathcal{Y}_t||_F^2 \\ & + w \sum_{r \in \mathbb{C}^{\mathcal{D}}} \sum_{t \in \mathbb{T}^{\mathcal{D}}} \rho_{r,r^*} \cdot ||\mathbf{x}_{r,t}^{rep} - \mathbf{x}_{r^*,t}^{rep}||^2 \end{aligned}$$

  epoch ++
end while
$\theta_{\mathcal{D}} \leftarrow \theta$
return $\theta_{\mathcal{D}}$

5 Experiment: Crowd Flow Prediction

We use crowd flow prediction as a case of spatio-temporal prediction tasks to evaluate *RegionTrans*.

5.1 Settings

Datasets. Following previous studies on crowd flow [Hoang *et al.*, 2016; Zhang *et al.*, 2017; Zhang *et al.*, 2016], we use bike flow data for evaluation. Three bike flow datasets collected from *Washington D.C.*, *Chicago*, and *New York City* are used. Each dataset covers a two-year period (2015-2016). In all the cities, the center area of $20km \times 20km$ are selected. The area is split to 20×20 regions (each region is $1km \times 1km$). For each evaluation scenario, we choose one city as the source city and another as the target. We assume that the source city has all its historical crowd flow data, but only limited crowd flow data exists in the target city (e.g., one day). The last two-month data is chosen for testing.

Metric. The evaluation metric is root mean square error (RMSE). Same as [Zhang *et al.*, 2017], the reported RMSE is the average RMSE of inflow and outflow.

Network Implementation. Our network structure implemented in the experiment has two layers of ConvLSTM with 5×5 filters and 32 hidden states, to generate $\mathbf{X}_t^{rep} \in \mathbb{R}^{20 \times 20 \times 32}$. With $\mathbf{X}_t^{rep}$ as the input, there is one layer of $\text{Conv2D}_{1\times1}$ with 32 hidden states, followed by another layer of $\text{Conv2D}_{1\times1}$ linking to the output crowd flow prediction. For the external context factors, e.g., temperature, wind speed, weather, and day type, we use the same feature extrac-

	D.C.→Chicago		Chicago→D.C.		D.C.→NYC		NYC→D.C.	
	1-day	3-day	1-day	3-day	1-day	3-day	1-day	3-day
Target Only								
ARIMA	0.740	0.694	0.707	0.661	0.360	0.341	0.707	0.661
DeepST	0.771	0.711	1.075	0.767	0.350	0.359	1.075	0.767
ST-ResNet	0.914	0.703	0.869	0.738	0.376	0.349	0.869	0.738
Source & Target								
DeepST (FT)	0.652	0.611	0.672	0.619	0.363	0.369	0.713	0.711
ST-ResNet (FT)	0.667	0.615	0.695	0.623	0.385	0.349	0.696	0.691
RegionTrans (S-Match)	0.605	0.594	0.631	0.602	**0.328**	**0.305**	**0.665**	**0.593**
RegionTrans (A-Match)	**0.587**	**0.576**	**0.600**	**0.581**	/	/	/	/

Table 1: Evaluation results. The target city holds 1 or 3-day crowd flow historical data. *RegionTrans (A-Match)* is available for D.C. $\rightleftharpoons$ Chicago as we have collected check-in data for Chicago and D.C.

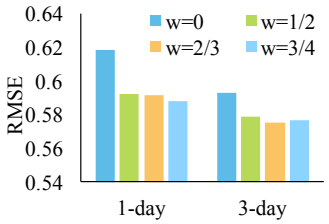

Figure 4: Tunning w (A-Match, D.C. $\rightarrow$ Chicago).

tion method as [Zhang *et al.*, 2017] and obtain an external feature vector with a length of 28. We also need to set w in Eq. 14 to balance the optimization trade-off between representation difference and prediction error. We set w to 0.75 as the default value. *ADAM* is used as the optimization algorithm [Kingma and Ba, 2015].

5.2 Methods

For *RegionTrans*, we implement two variants:

- *RegionTrans (S-Match)*: learning the inter-city region matching function $\mathcal{M}$ only by the short period of the target city *Service* data, i.e., crowd flow.
- *RegionTrans (A-Match)*: learning the inter-city region matching function $\mathcal{M}$ by the long period of the *Auxiliary* data, i.e., Foursquare check-in data. We use one-year check-in data as the auxiliary data since it is a useful indication of crowd flow [Yang *et al.*, 2016]. Note that we have collected check-in data from D.C. and Chicago, so *RegionTrans (A-Match)* is available for the knowledge transfer between them.

We compare *RegionTrans* with two types of baselines. The first type only uses the short crowd data history of target city:

- *ARIMA*: a widely-used time series prediction method in statistics [Hyndman and Athanasopoulos, 2014].
- *DeepST* [Zhang *et al.*, 2016]: a deep spatio-temporal neural network based on convolutional network. The complete DeepST model has three components: *closeness*, *period*, and *trend*. But the *period* and *trend* components can only be activated if the training data last for more than one day and seven days, respectively. Therefore, if the target city does not have enough data, we have to deactivate the corresponding components.
- *ST-ResNet* [Zhang *et al.*, 2017]: a deep spatio-temporal network based on residual network [He *et al.*, 2016].

The second type trains a deep model on the source city data, and *fine-tune* it with the target city data:

- *DeepST (FT)*: fine-tuned DeepST.
- *ST-ResNet (FT)*: fine-tuned ST-ResNet.

As mentioned in Sec. 4.3, DeepST and ST-ResNet predict the city crowd flow as a whole, and thus we cannot fine tune their models between two cities of different sizes. Therefore, to make the comparison possible, our experiment selects the same area size for two cities. Note that *RegionTrans* is able to transfer knowledge between two cities of different sizes, and thus is more flexible.

5.3 Results

Table 1 shows our results for D.C. $\rightleftharpoons$ Chicago and D.C. $\rightleftharpoons$ NYC. *RegionTrans* can consistently outperform the best baseline, where the largest improvement is reducing RMSE by up to 10.7%. In particular, when the service data of the target city is smaller, the improvement of *RegionTrans* is usually more significant. This indicates that the inter-city similar-region pairs are valuable for transfer learning especially when target data is extremely scarce. Between two *RegionTrans* variants, *RegionTrans (A-Match)* is better, as shown in D.C. $\rightleftharpoons$ Chicago. This implies that if an appropriate type of auxiliary data exists, it is possible to build a better inter-city region matching than only using the short period of the service data. If the auxiliary data is unavailable, using the limited period of service data for region matching can still lead to the competitive variant *RegionTrans (S-Match)*, which beats all the baselines significantly.

We also train DeepST with the full training data in Chicago and D.C., respectively, leading to RMSE of 0.521 and 0.526, which could be seen as the performance upper bound of our experiment cases. Compared to *RegionTrans* (DC→Chicago, A-Match, 3-day) result of 0.576, the full Chicago model reduces 9.6% of RMSE; the full DC model is better than *RegionTrans* (Chicago→DC, A-Match, 3-day) by reducing 9.5% of RMSE. In comparison, the best baseline DeepST (FT, 3-day) is worse than the full model by 14.7% and 15.0% in Chicago and DC, respectively. Hence, *RegionTrans* can reduce the gap between the best baseline and the upper bound full model by $\sim 1/3$, which is a significant improvement.

Another important observation is that *RegionTrans* is more robust when transferring knowledge between two dissimilar cities than baselines. Between the three cities in the experiment, D.C. and Chicago are similar in population, while NYC has a much larger population. This indicates that the knowledge transfer between D.C. $\rightleftharpoons$ Chicago may be easier, while D.C. $\rightleftharpoons$ NYC could be harder. Our results also verify this as DeepST and ST-ResNet get large improvement by fine-tuning in D.C. $\rightleftharpoons$ Chicago; but in D.C. $\rightarrow$ NYC, *negative trans-*

fer appears for the fine-tuned DeepST and ST-ResNet, leading to even worse performance than ARIMA, indicating that directly transferring the whole city knowledge from D.C. to NYC is ineffective. In comparison, *RegionTrans* consistently achieves lower error than all the baselines, verifying that the knowledge from D.C. can still be effectively transferred to NYC. The primary reason that *RegionTrans* can avoid negative transfer is that although D.C. and NYC are dissimilar in general, we can still find inter-city region pairs with similar spatio-temporal patterns (e.g., central business district).

Tuning w. We tune w in Eq. 14 to see how it will affect the performance. The larger w is, the higher weight is put on minimizing the similar-region representation difference. Fig. 4 shows the results. If we set $w = 0$ (i.e., direct fine tuning without region matching), the performance is significantly worse than when $w > 0$, by incurring up to 5% higher error. This highlights the effectiveness of our proposed inter-city similar-region matching scheme in cross-city knowledge transfer. For other settings of $w > 0$, the performance difference is minor. A larger w performs slightly better for a very short period of target city crowd flow data, e.g., one day.

Computation time. We use a server with Intel Xeon CPU E5-2650L, 128 GB RAM, and Nvidia Tesla M60 GPU. We implement *RegionTrans* with TensorFlow (CentOS). Training the source city model on two-year data needs $\sim$20 minutes, and the transfer learning for the target city model costs $\sim$50 and $\sim$100 minutes for 1 and 3-day data, respectively. This running time efficiency is acceptable in real-life deployments.

6 Conclusion

To address the data scarcity issue in urban spatio-temporal prediction tasks, this paper proposes a novel cross-city deep transfer learning method, called *RegionTrans*. Note that while this paper focuses on cross-city transfer, the key idea of *RegionTrans* can be flexibly leveraged with other spatial granularities, e.g., district-level or country-level transfer. In the future, we will study how effective transfer can be done on other recent spatio-temporal models such as *graph* convolutional networks [Geng *et al.*, 2019; Li *et al.*, 2018].

References

[Geng *et al.*, 2019] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Jieping Ye, Yan Liu, and Qiang Yang. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *AAAI*, 2019.

[Guo *et al.*, 2018] Bin Guo, Jing Li, Vincent W. Zheng, Zhu Wang, and Zhiwen Yu. Citytransfer: Transferring inter- and intra-city knowledge for chain store site recommendation based on multi-source urban data. *ACM IMWUT*, 1(4):135:1–135:23, January 2018.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Hoang *et al.*, 2016] Minh X Hoang, Yu Zheng, and Ambuj K Singh. Fccf: forecasting citywide crowd flows based on big data. In *SIGSPATIAL*, page 6, 2016.

[Hyndman and Athanasopoulos, 2014] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2014.

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

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Li *et al.*, 2018] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *ICLR*, 2018.

[Lin *et al.*, 2014] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *ICLR*, 2014.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE TKDE*, 22(10):1345–1359, 2010.

[Shi *et al.*, 2015] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, pages 802–810, 2015.

[Wang *et al.*, 2017] Leye Wang, Daqing Zhang, Dingqi Yang, Animesh Pathak, Chao Chen, Xiao Han, Haoyi Xiong, and Yasha Wang. Space-ta: Cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing. *ACM TIST*, 9(2):20, 2017.

[Wang *et al.*, 2019] Leye Wang, Xu Geng, Xiaojuan Ma, Daqing Zhang, and Qiang Yang. Ridesharing car detection by transfer learning. *Artificial Intelligence*, 273:1–18, 2019.

[Wei *et al.*, 2016] Ying Wei, Yu Zheng, and Qiang Yang. Transfer knowledge between cities. In *KDD*, pages 1905–1914, 2016.

[Yang *et al.*, 2016] Dingqi Yang, Daqing Zhang, and Bingqing Qu. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM TIST*, 7(3):30, 2016.

[Yao *et al.*, 2018] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, and Jieping Ye. Deep multi-view spatial-temporal network for taxi demand prediction. In *AAAI*, 2018.

[Zhang *et al.*, 2016] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. Dnn-based prediction model for spatio-temporal data. In *SIGSPATIAL*, page 92, 2016.

[Zhang *et al.*, 2017] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI*, pages 1655–1661, 2017.

[Zheng *et al.*, 2014] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: concepts, methodologies, and applications. *ACM TIST*, 5(3):38, 2014.