Crowd Flow Prediction by Deep Spatio-Temporal Transfer Learning

Leye Wang
The Hong Kong University of Science and Technology
72 PUBLICATIONS **1,597** CITATIONS
SEE PROFILE

Xu Geng
The Hong Kong University of Science and Technology
5 PUBLICATIONS **54** CITATIONS
SEE PROFILE

Xiaojuan Ma
Huawei Technologies
33 PUBLICATIONS **558** CITATIONS
SEE PROFILE

Crowd Flow Prediction by Deep Spatio-Temporal Transfer Learning

Leye Wang[1*], Xu Geng[1*], Xiaojuan Ma[1], Feng Liu[2], Qiang Yang[1]
[1] Hong Kong University of Science and Technology
[2] SAIC Motor Corporation Limited
wly@cse.ust.hk, xgeng@connect.ust.hk, mxj@cse.ust.hk, liufeng@saicmotor.com, qyang@cse.ust.hk
[*] equal contribution

Abstract

Crowd flow prediction is a fundamental urban computing problem. Recently, deep learning has been successfully applied to solve this problem, but it relies on rich historical data. In reality, many cities may suffer from data scarcity issue when their targeted service or infrastructure is new. To overcome this issue, this paper proposes a novel deep spatio-temporal transfer learning framework, called *RegionTrans*, which can predict future crowd flow in a data-scarce (target) city by transferring knowledge from a data-rich (source) city. Leveraging social network check-ins, *RegionTrans* first links a region in the target city to certain regions in the source city, expecting that these inter-city region pairs will share similar crowd flow dynamics. Then, we propose a deep spatio-temporal neural network structure, in which a hidden layer is dedicated to keeping the *region representation*. A source city model is then trained on its rich historical data with this network structure. Finally, we propose a region-based cross-city transfer learning algorithm to learn the target city model from the source city model by minimizing the hidden representation discrepancy between the inter-city region pairs previously linked by check-ins. With experiments on real crowd flow, *RegionTrans* can outperform state-of-the-arts by reducing up to 10.7% prediction error.

1 Introduction

Predicting future crowd flow (i.e., the amount of inflow and outflow of pedestrian, taxi, bus, bike, etc.) is a fundamental problem in urban computing. For city planners, this problem is critical to traffic management and public safety issues [Zhang *et al.*, 2017]. For companies like Uber and DiDi, accurate crowd flow prediction helps design better business strategies to balance driver supply and passenger demand.

Recently, researchers begin to apply deep learning to the crowd flow prediction problem, and verify that deep leaning can outperform traditional machine learning and statistic methods [Zhang *et al.*, 2016; Zhang *et al.*, 2017]. However, existing deep crowd flow prediction methods need a long record of past crowd flow data for training, and many cities may not meet this requirement in reality. For example, local government may just start urban digitalization process and do not have many historical data stored, or a company opens its business in a new city. In such cases, it is hard to build a robust deep prediction model only with the target city's own historical crowd flow data.

In this paper, we propose a deep transfer learning framework, *RegionTrans*, to predict future crowd flow in a data-scarce city (*target city*) by transferring knowledge of crowd flow dynamic patterns learned from a data-rich city (*source city*) at a *region* level. The principle idea is to find *inter-city region pairs* that share similar crowd flow dynamic patterns and then use such region pairs as proxies to efficiently transfer knowledge from source city to the target. To achieve this goal, we face two challenges:

(i) As there is few crowd flow data in the target city, we may not be able to directly compute a reliable crowd flow similarity between a region in the source city (*source region*) and a region in the target city (*target region*). Then, how can we find similar inter-city region pairs robustly?

(ii) As existing deep learning approaches are often designed to predict citywide crowd flow as a whole, it is hard to incorporate region-level knowledge into them. Then, how can we leverage the inter-city region similarity information for effective deep transfer learning?

To address the first issue, we turn to the social network check-in data widely available across the world. Intuitively, if two regions have similar check-in patterns, their crowd flow dynamic patterns might also be similar. In other words, there may be implicit relationship between check-ins and crowd flows. With this intuition, we can model a check-in feature representation to measure the similarity between source and target regions, even though target regions have few crowd flow data. Then, for each target region, we match it to its top-k most similar source regions to construct a set of inter-city similar-region pairs, which will later be used to facilitate knowledge transfer.

To deal with the second issue, we propose a new deep spatio-temporal neural network structure and a corresponding region-based cross-city transfer learning algorithm. Compared to existing deep spatio-temporal networks [Zhang *et al.*, 2017; Zhang *et al.*, 2016], the key novelty of our network structure is its capability to output *region representations* in a hidden layer (i.e., a latent feature for each region). This allows us to encode inter-city region similarity information into this layer for knowledge transfer. More specifically, we first train a deep crowd flow prediction model in the source city with its rich historical data. Then, with few historical data in the target city, we optimize this model by minimizing not only the prediction error, but also the discrepancy of the latent feature representations between inter-city similar-regions. This ensures that the latent feature representation of each target region will be close to that of its corresponding source regions, so as to boost the knowledge transfer performance.

Briefly, this paper has the following contributions.

(i) To the best of our knowledge, this is the first work to study how to facilitate crowd flow prediction in a data-scarce city (target city) by transferring knowledge from a data-rich city (source city) via deep learning.

(ii) We propose a novel deep transfer learning framework called *RegionTrans*. We first learn a check-in feature from social network data which correlates with the crowd flow dynamics of a region. Using this check-in feature, for each target region, we can find its top-k similar source regions and form a set of inter-city similar-region pairs. Then, we construct a novel deep spatio-temporal neural network structure with hidden region representations. Finally, we propose a region-based cross-city transfer learning algorithm to learn a deep crowd flow prediction model for the target city by considering both the inter-city similar-region pairs and the deep model of the source city.

(iii) Evaluations on real crowd flow dataset have shown the effectiveness of RegionTrans. Compared to fine-tuned state-of-the-art deep spatio-temporal crowd flow prediction methods, RegionTrans can reduce up to 10.7% prediction error.

2 Problem Formulation

In this section, we first define some key concepts, and then formulate the problem of predicting the crowd flow of a data-scarce city by transfer learning from a data-rich city.

Definition 1. Region [Zhang *et al.*, 2016]. A city is partitioned into $W \times H$ 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 is denoted as $\mathbb{C}$.

Definition 2. Crowd flow [Zhang *et al.*, 2016]. We have two types of crowd flows: *inflow* and *outflow*. Inflow of a region r at k^{th} time interval, denoted as $\mathcal{I}_{r,k}$, is the number of objects (e.g., cars and pedestrians) which are in r at k^{th} time interval but outside of r at $k-1^{th}$ time interval. Outflow of r at k^{th} time interval, denoted as $\mathcal{O}_{r,k}$, is the number of objects which are outside of r at k^{th} time interval but inside r at $k-1^{th}$ time interval.

For brevity, we only consider equal-length time interval (e.g., one-hour) as in the previous research [Zhang *et al.*, 2016; Zhang *et al.*, 2017].

Problem. Suppose that the current time interval is t and there is a source city sc with a long historical record of crowd flow data lasting for T_{sc} time intervals:

$$\{(\mathcal{I}^{sc}_{r,k}, \mathcal{O}^{sc}_{r,k}) | r \in \mathbb{C}_{sc}, k \in [t - T_{sc} + 1, t]\} \tag{1}$$

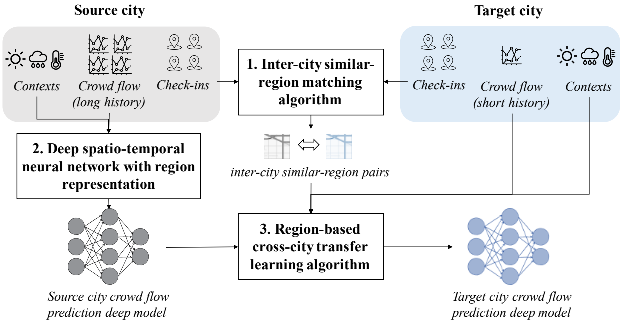

Source city
Target city
Contexts
Crowd flow (long history)
Check-ins
1. Inter-city similar-region matching algorithm
Check-ins
Crowd flow (short history)
Contexts
2. Deep spatio-temporal neural network with region representation
inter-city similar-region pairs
3. Region-based cross-city transfer learning algorithm
Source city crowd flow prediction deep model
Target city crowd flow prediction deep model

Figure 1: Overview of RegionTrans.

There is another target city tc with a short historical record of crowd flow data lasting for $T_{tc} (\ll T_{sc})$ time intervals:

$$\{(\mathcal{I}^{tc}_{r,k}, \mathcal{O}^{tc}_{r,k}) | r \in \mathbb{C}_{tc}, k \in [t - T_{tc} + 1, t]\} \tag{2}$$

The problem is to predict the crowd flow of tc in the next time interval $t + 1$:

$$\{(\mathcal{I}^{tc}_{r,t+1}, \mathcal{O}^{tc}_{r,t+1}) | r \in \mathbb{C}_{tc}\} \tag{3}$$

3 Methodology

To solve the above problem, we propose a deep transfer learning framework *RegionTrans*, which will be elaborated next.

3.1 Framework Overview

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

(i) **Inter-city similar-region matching algorithm from social check-in data**. As the target city only has a short history of crowd flow, directly calculating the similarity between a source region and a target region using flow data may not yield robust results. For example, suppose that 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 similarity is inadequate. To address this issue, we then rely on the social network check-ins in both source and target cities to calculate the inter-city region similarity. Our intuition is that if the check-in patterns of two regions are similar, their crowd flow dynamic patterns may be similar. As check-in data are widely and openly available, we can get a much longer history of data so as to ensure the check-in similarity measurement is more reliable.

(ii) **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 crowd flow prediction model from its long historical record of crowd flow and corresponding contexts (e.g., weather). This source city model will be later used in transfer learning for building the target city model.

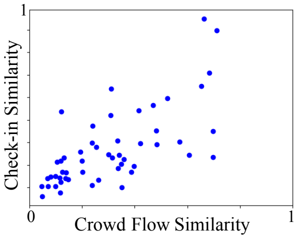


Figure 2: Inter-city region check-in and crowd flow similarities.

(iii) **Region-based cross-city transfer learning algorithm**. Based on the deep model with region representations, we propose a transfer learning 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.

3.2 Inter-city Similar-region Matching

To robustly measure the similarity between inter-city regions, we turn to the widely accessible social media check-in data. For a certain region $r \in \mathbb{C}$ in a city, we model the check-in representation according to its hourly check-in counts in workday and weekend/holiday as follows:

$$\boldsymbol{ch_r} = \langle ch_0, ch_1, \cdots, ch_{23}, ch'_0, ch'_1, \cdots, ch'_{23}\rangle, r \in \mathbb{C} \tag{4}$$

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.

With this representation, for each region of target city tc, we then identify the top k regions of source city sc that have the most similar check-in patterns. More specifically, we measure the check-in similarity between regions using the *Pearson* correlation coefficient. We denote the set of similar regions of a target city region r as $\mathcal{M}(r)$:

$$\mathcal{M}(r) = \{r^*_1, \cdots, r^*_k\},\ r \in \mathbb{C}_{tc}, r^*_1, \cdots, r^*_k \in \mathbb{C}_{sc} \tag{5}$$

$$\rho_{r,r^*} \geq \rho_{r,r'}, \qquad r^* \in \mathcal{M}(r), r' \in \mathbb{C}_{sc} \setminus \mathcal{M}(r) \tag{6}$$

$$\rho_{r,r^*} = Pearson(\boldsymbol{ch_r}, \boldsymbol{ch_{r^*}}) \tag{7}$$

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 (details of the dataset are shown in the evaluation section). Here, we measure the crowd flow similarity between two regions by first counting the hourly inflow/outflow counts and then use the Pearson correlation coefficient. Figure 2 plots both the check-in similarity (y axis) and crowd flow similarity (x axis) of each D.C. region (blue point) with a selected Chicago region. From the figure, we can observe that the D.C. regions with higher check-in similarities tend to hold higher crowd flow similarities, which verifies the effectiveness of our inter-city similar-region matching method.

3.3 Deep Spatio-temporal Neural Network with Region Representations

Figure 3 shows our network structure for citywide crowd flow prediction. We first describe the crowd flow input and output. Second, we illustrate the structure of our network in detail.

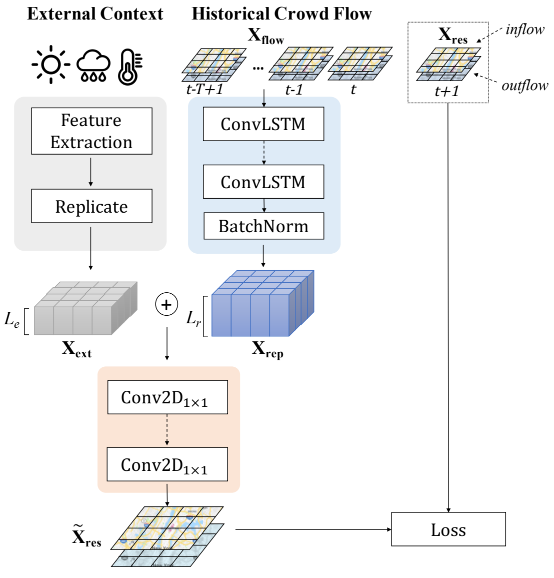


Figure 3: Our network structure.

Finally, we elaborate the most important part of our network structure, i.e., region representations.

Crowd flow input and output. As pointed by Definition 1, we convert a city map into $W \times H$ regions. Suppose the historical crowd flow lasts for T cycles, then the input crowd flow $\mathbf{X}_{flow}$ is a tensor $\in \mathbb{R}^{T \times W \times H \times 2}$; each 2-length vector $\mathbf{X}_{flow}[t, w, h, :]$ represents the value of the inflow and outflow of the region $r_{[w,h]}$ at the time interval t. The output of our neural network, denoted as $\widetilde{\mathbf{X}}_{res} \in \mathbb{R}^{W \times H \times 2}$, is the predicted citywide crowd flow in the next time interval. The objective of our neural network is to minimize the mean squared error between $\widetilde{\mathbf{X}}_{res}$ and the real crowd flow $\mathbf{X}_{res}$ in the next time interval:

$$\arg\min_{\theta} ||\widetilde{\mathbf{X}}_{res} - \mathbf{X}_{res}||_2^2 \tag{8}$$

where θ is the set of network parameters.

Network structure. The basic components of our network are convolutional LSTM (ConvLSTM) layers [Shi *et al.*, 2015]. ConvLSTM is a variant of LSTM by replacing dense kernels with convolution ones. It is able to capture both spatial and temporal dependencies within the data. Our neural network leverages ConvLSTM to construct hidden feature to catch both spatial and temporal patterns of crowd flow. Briefly, a ConvLSTM layer can map a time sequence of input $\in \mathbb{R}^{T \times W \times H \times 2}$ to a time sequence of output $\in \mathbb{R}^{T \times W \times H \times L}$, where L is the number of hidden states. After stacking several layers of ConvLSTM, for the last ConvLSTM layer, we keep the last time interval in the output, leading to an output $\in \mathbb{R}^{W \times H \times L_r}$. With batch normalization, this hidden layer output, denoted as $\mathbf{X}_{rep}$, is a key part of our structure, which actually can keep region representations (i.e., each region $r_{[w,h]}$ has a L_r-length representation vector $\mathbf{X}_{rep}[w, h, :]$). We will discuss it later in more details.

After getting $\mathbf{X}_{rep}$, we incorporate the external context factors into the network structure. External context factors include temperature, weather, wind speed, day type (workday/holiday), etc., which will also impact the crowd flow dy-

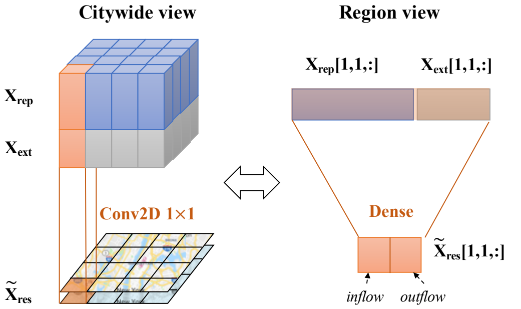

Citywide view
Region view
$\mathbf{X}_{\text{rep}}$
$\mathbf{X}_{\text{ext}}$
Conv2D 1×1
$\widetilde{\mathbf{X}}_{\text{res}}$
$\mathbf{X}_{\text{rep}}[1,1,:]$
$\mathbf{X}_{\text{ext}}[1,1,:]$
Dense
$\widetilde{\mathbf{X}}_{\text{res}}[1,1,:]$
inflow
outflow

Figure 4: Citywide and region view of our neural network structure.

namics. Same as [Zhang *et al.*, 2017], we encode the external factors of the next time interval $t+1$ into a L_e-length feature vector[1], and replicate it for all the regions to get a citywide external representation $\mathbf{X}_{ext} \in \mathbb{R}^{W \times H \times L_e}$. By concatenating $\mathbf{X}_{rep}$ and $\mathbf{X}_{ext}$ to form a representation $\in \mathbb{R}^{W \times H \times (L_r + L_e)}$, we finally employ several convolution 2D layers with 1×1 filters (Conv2D$_{1\times1}$, first introduced in [Lin *et al.*, 2014]) to predict the next-time-interval crowd flow $\widetilde{\mathbf{X}}_{res} \in \mathbb{R}^{W \times H \times 2}$. In fact, why we use Conv2D$_{1\times1}$ layers here is to make $\mathbf{X}_{rep}$ be able to reflect region representation, as discussed below.

Region representation. Now we elaborate the key novel component of our proposed neural network structure, i.e., the region representation $\mathbf{X}_{rep}$. First, we explain why $\mathbf{X}_{rep}$ can be seen as region representation. As we use Conv2D$_{1\times1}$ layers after we combining $\mathbf{X}_{rep}$ and $\mathbf{X}_{ext}$, the final output predicted crowd flow in a region will only be affected by the corresponding L_r-length vector in $\mathbf{X}_{rep}$. For example, the predicted inflow and outflow of region $r_{[1,1]}$, i.e., 2-length vector $\widetilde{\mathbf{X}}_{res}[1,1,:]$, will only be affected by the L_r-length vector $\mathbf{X}_{rep}[1,1,:]$. In other words, from the perspective of the single region $r_{[1,1]}$, we can see $\mathbf{X}_{rep}[1,1,:]$, after concatenating the L_e-length external feature vector, connects to several dense layers and then outputs the predicted crowd flow for region $r_{[1,1]}$. Therefore, conceptually we can see $\mathbf{X}_{rep}[1,1,:]$ as a representation of $r_{[1,1]}$ to reflect its crowd flow dynamics. Figure 4 visualizes the above explanation.

We note that such a region representation design is different from previous work [Zhang *et al.*, 2017; Zhang *et al.*, 2016] where the citywide crowd flow is seen as a whole. Although previous methods can also be used in cross-city transfer learning with mechanisms such as fine tunning, we highlight that our neural network structure with region representations has several distinct advantages for cross-city knowledge transfer.

(i) *Fine-grained region-level transfer*. With previous methods which see citywide crowd flow as a whole, we can only transfer the knowledge from the whole source city to the target city. If two cities are not similar, the transfer performance may be poor. However, with region representation, we can make fine-grained knowledge transfer based on region similarity. As long as we can find similar region pairs between cities, the effective transfer may be conducted.

Algorithm 1 Region-based cross-city transfer learning

Input:
θ_{sc}: Pre-trained network parameters on source city with a long period of data
TR_{tc}: target city training data (a short time period T^*)
TR_{sc}: source city training data (a short time period T^*)
$\mathcal{M}$: inter-city similar-region matching scheme
Output:
θ_{tc}: network parameters for the target city

```
Initialize network parameters: $\theta \leftarrow \theta_{sc}$
epoch $\leftarrow 0$
while epoch $\leq$ MAX_EPOCH do
    for each $\{\mathbf{X}_{flow}, \mathbf{X}_{ext}, \mathbf{X}_{res}\} \in TR_{tc}$ do
        Get corresponding $\{\mathbf{X}'_{flow}, \mathbf{X}'_{ext}, \mathbf{X}'_{res}\} \in TR_{sc}$ (same time span)
        $\mathbf{X}'_{rep} \leftarrow$ region representation of network ($\theta$) with input $\mathbf{X}'_{flow}, \mathbf{X}'_{ext}$
        for $i \in [1, k]$ do
            $\hat{\mathbf{X}}^i_{rep} \leftarrow \mathbf{0}^{W \times H \times L_r}$ (a tensor with all zeros)
            for $r_{[w,h]} \in \mathbb{C}_{tc}$ do
                $r'_{[w',h']} \leftarrow \mathcal{M}(r_{[w,h]})[i]$ (note that $r'_{[w',h']} \in \mathbb{C}_{sc}$)
                $\hat{\mathbf{X}}^i_{rep}[w, h, :] \leftarrow \mathbf{X}'_{rep}[w', h', :]$
            end for
        end for
        $\theta \leftarrow \arg\min_{\theta} \; w(\frac{1}{k} \sum_{1 \leq i \leq k} ||\boldsymbol{\rho}_i \circ (\mathbf{X}_{rep} - \hat{\mathbf{X}}^i_{rep})||_2^2)$
                                    $+ (1-w)||\widetilde{\mathbf{X}}_{res} - \mathbf{X}_{res}||_2^2$
    end for
    epoch ++
end while
$\theta_{tc} \leftarrow \theta$
return $\theta_{tc}$
```

(ii) *Transfer between cities with different sizes*. Since our neural network structure can actually be seen from region view (Figure 4), even if two cities have different sizes (i.e., different $W \times H$ in Def. 1), 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 previous network structures [Zhang *et al.*, 2017; Zhang *et al.*, 2016], if we want to transfer a learned model from the source city to the target one, the two cities have to be the same size.

3.4 Region-based Cross-City Transfer Learning

With our neural network structure, we can first train a deep crowd flow prediction model in the source city with its rich history of crowd flow data (e.g., several months). We denote θ_{sc} as the network parameters learned from the source city. Then, with θ_{sc} as the pre-trained network parameters, we propose a region-based cross-city transfer learning algorithm to further optimize the parameters to improve its performance on the target city, considering a short period T^* (e.g., only a few days) of the crowd flow data in the target city and the inter-city similar-region matching scheme $\mathcal{M}$. The detailed algorithm is shown in Algorithm 1.

The principle idea of our transfer learning algorithm is when optimizing the network parameters θ, we not only make the predicted crowd flow close to the true crowd flow in the training data during T^* on the target city, but also let the hidden representation $\mathbf{X}_{rep}$ of a target region r be close to the representations of its top-k similar-regions in the source city, i.e., $\mathcal{M}(r)$. More specifically, if a target region and a source region have a higher check-in similarity, we put a higher weight on minimizing the difference between their represen-

[1] Some external factors, like weekday/holiday of the time interval $t+1$ can be directly obtained. The others, like weather, can be set to the value of time interval t for approximation as the time interval length is usually short (e.g., 30 minutes) so that these values will not change significantly in general [Zhang *et al.*, 2017].

tations. Suppose the target city has the map size of $W \times H$, then we can write the optimization objective as follows:

$$\begin{aligned}\arg\min_{\theta} \quad & w(\frac{1}{k}\sum_{1 \le i \le k} ||\boldsymbol{\rho}_i \circ (\mathbf{X}_{rep} - \hat{\mathbf{X}}^i_{rep})||_2^2) \\ & + (1-w)||\widetilde{\mathbf{X}}_{res} - \mathbf{X}_{res}||_2^2\end{aligned} \tag{9}$$

where $\circ$ is element-wise multiplication; $\hat{\mathbf{X}}^i_{rep} \in \mathbb{R}^{W \times H \times L_r}$, and $\hat{\mathbf{X}}^i_{rep}[w, h, :]$ is the hidden representation of the i^{th} similar-region of the target region $r_{[w,h]}$; $\boldsymbol{\rho}_i \in \mathbb{R}^{W \times H}$ is a matrix where each element stores the check-in Pearson correlation coefficient of a target region and its i^{th} similar-region; w is the weight to trade off between minimizing the representation difference or minimizing the prediction error. The transfer learning process continues until when the parameters converge or iterations reach the maximum number.

4 Experiments

4.1 Settings

Datasets. Following previous studies [Hoang *et al.*, 2016; Zhang *et al.*, 2016; Zhang *et al.*, 2017], we use bike flow as a case of crowd flow for evaluation. Two bike flow datasets collected from *Washington D.C.* and *Chicago* are used. Each dataset covers a two-year period (2015-2016). In both cities, the center area of $20km \times 20km$ are selected as the studied area. The area is split to 20 × 20 regions (i.e., each region is $1km \times 1km$). Social network check-in data come from Foursquare [Yang *et al.*, 2016]. Weather data are from OpenWeatherMap. We summarize the dataset statistics in Table 1.

Scenarios. We evaluate two scenarios: using Washington D.C. as the source city and Chicago as the target, and vice versa. In each scenario, we assume that the source city has all its historical crowd flow data, but only a very limited period of historical crowd flow data exists in the target city (e.g., one day). The last two month data are chosen for testing. 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 $\mathbf{X}_{rep} \in \mathbb{R}^{20 \times 20 \times 32}$. With $\mathbf{X}_{rep}$ as the input, there is one layer of Conv2D$_{1\times1}$ with 32 hidden states, followed by another layer of Conv2D$_{1\times1}$ linking to the output crowd flow prediction. For the external factors, e.g., temperature, wind speed, weather, and day type, we use the same feature extraction method as [Zhang *et al.*, 2017] and obtain an external feature vector with a length of 28.

Parameters. RegionTrans has two parameters to set. The first is k in top-k similar-regions detected by the inter-city similar-region matching algorithm. We set k to 5 as the default value. The second is w in Eq. 9, which is used to balance the optimization trade-off between representation difference and prediction error. We set w to 0.75 as the default value.

Baselines. We compare RegionTrans to two types of baselines. The first type only uses the short crowd data history of target city for training its prediction model:

	Washington D.C.	**Chicago**
#Trip records	6,519,741	6,690,351
Time span	2015.1.1 - 2016.12.31	
Time interval	30 minutes	
Region size	$1km \times 1km$	
City map size	20×20	

Table 1: Dataset statistics.

	D.C.→Chicago			**Chicago→D.C.**		
	1-day	3-day	7-day	1-day	3-day	7-day
Target Data Only						
ARIMA	0.740	0.694	0.679	0.707	0.661	0.647
DeepST	0.771	0.711	0.636	1.075	0.767	0.691
ST-ResNet	0.914	0.703	1.053	0.869	0.738	1.054
Source & Target Data						
DeepST (FT)	0.652	0.611	0.566	0.672	0.619	0.586
ST-ResNet (FT)	0.667	0.615	0.613	0.695	0.623	0.608
RegionTrans	**0.587**	**0.576**	**0.553**	**0.600**	**0.581**	**0.573**

Table 2: Evaluation results.

- *ARIMA*: Auto-Regressive Integrated Moving Average is a widely-used time series prediction method.
- *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 neural network based on residual network [He *et al.*, 2016]. Same as DeepST, ST-ResNet has three components. We then adapt ST-ResNet in the same way as DeepST in our experiments.

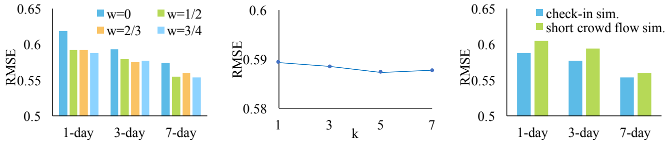
The second type of baselines first trains a deep model on the source city data, and *fine-tune* it with the target city data:

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

As mentioned in Sec. 3.3, 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 in the two cities. Note that RegionTrans is able to transfer knowledge between two cities of different sizes, and thus is more flexible than DeepST and ST-ResNet.

4.2 Results

Comparison with baselines. Table 2 shows our evaluation results. In both scenarios, RegionTrans can consistently outperform the best baseline by reducing the prediction error by 2.2%-10.7%. In particular, when the recorded history of the target city is shorter, the improvement of RegionTrans is more significant. This indicates that the introduced intercity similar-region pairs are valuable for transfer learning especially when target data are extremely scarce. Among the baselines, we observe that the deep models, i.e., DeepST and ST-Resnet with only target data for training, perform rather poorly and unstably, often worse than ARIMA. Fine-tuned



(a) vary w (b) vary k (one-day) (c) check-in effect

Figure 5: Other results of RegionTrans (D.C. → Chicago).

DeepST and ST-Resnet are much better, as the pre-trained model on the source city gives a good start point for optimizing network parameters. RegionTrans, by further considering the inter-city region similarity information in transfer learning, is able to outperform the two fine-tuned deep models.

Tuning w. Here, we tune w in Eq. 9 to see how it will affect the performance. The larger w is, the higher weight is put on minimizing the similar-region representation difference. Figure 5a shows the results. If we set $w = 0$, i.e., ignoring the inter-city similar-region representation in transfer learning, 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 when we have a very short period of target city crowd flow data, e.g., one day of record.

Tuning k. In the inter-city similar-region matching step, for each target region, we link it to the top-k similar source regions. By tuning k, we consider different numbers of similar source regions in transfer learning. The prediction error of RegionTrans of different settings of k is shown in Figure 5b. Generally, different settings of k perform quite similarly.

Effect of check-in similarity. To verify the effectiveness of check-ins in matching inter-city similar regions, we change the matching criteria from check-in similarity to the short period crowd flow similarity (i.e., using 1/3/7-day crowd flow data to calculate inter-city region similarity and select top-k similar regions). Figure 5c shows results. If we use short crowd flow data for region matching, the error is higher than RegionTrans with check-in data for region matching. This verifies that the inter-city region check-in similarity is more reliable for our transfer learning task.

Computation time. The experiment platform is equipped with Intel Xeon CPU E5-2650L, 128 GB RAM, and Nvidia Tesla M60 GPU. We implement RegionTrans with TensorFlow in CentOS. Training the source city model on two years of data needs about 20 minutes, and the transfer learning for the target city model costs about 50, 100, and 160 minutes for 1, 3, 7-day data, respectively. This running time efficiency is acceptable in real-life deployments.

5 Related Work

Crowd flow prediction is a fundamental problem in urban computing [Zheng *et al.*, 2014]. Most studies on this topic predicts the traffic volumes in a single or multiple road segments or regions [Silva *et al.*, 2015; Wang *et al.*, 2017b]. Recently, researchers begin to take the whole city into consideration and predict the citywide crowd flow all together on various scenarios, e.g., taxi and bike flows [Li *et al.*, 2015; Chen *et al.*, 2016; Hoang *et al.*, 2016; Zhang *et al.*, 2016; Zhang *et al.*, 2017]. Inspired by the deep network structures proposed for spatio-temporal learning tasks such as precipitation nowcasting [Shi *et al.*, 2015] and future video frame prediction [Mathieu *et al.*, 2016], deep learning is also adopted in crowd flow prediction and becoming the state-of-the-art solution when there exists a rich history of crowd flow data. Various deep models have been used, e.g., CNN [Zhang *et al.*, 2016] and ResNet [Zhang *et al.*, 2017]. Compared to these works, the significant difference of our work lies in both the objective and the method. We aim to apply the deep model to a target city which only has a short history of crowd data, and thus propose RegionTrans to effectively transfer knowledge from a data-rich source city to the target city at the region level.

Transfer learning is adopted to address the machine learning problem when labeled and 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 strategy 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.*, 2017b], and leveraging the taxi GPS traces to detect ridesharing cars such as Uber [Wang *et al.*, 2017a]. The second strategy is to find a source city with adequate data to transfer knowledge to the target city. Guo *et al.* design a cross-city transfer learning framework with collaborative filtering and AutoEncoder to conduct chain store site recommendation in a new city [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 transfer learning algorithm FLORAL to predict air quality category in a target city by transferring knowledge from a source city. There are two difficulties to apply FLORAL to our task: (1) crowd flow prediction is a regression task but FLORAL is designed for classification; (2) FLORAL is not designed for deep learning. In brief, to the best of our knowledge, RegionTrans is the first deep spatio-temporal transfer learning framework that facilitates urban crowd flow prediction in a data-scarce target city.

6 Conclusion and Future Work

In this paper, to address the data scarcity issue in crowd flow prediction, we propose a novel deep spatio-temporal transfer learning framework, called *RegionTrans*. Our novelties lie in three aspects. (1) We use auxiliary data (i.e., check-ins) to obtain inter-city region similarities correlated to crowd flow dynamics. (2) We design a deep spatio-temporal model with a hidden layer dedicated to storing region latent representations. (3) We propose a learning algorithm to transfer knowledge from a source city to a target one by considering the latent representations of the inter-city similar-region pairs.

In the future, we plan to extend RegionTrans in several directions. First, we will further improve the reliability of intercity similar-region pairs by exploiting other sources of auxiliary data such as points-of-interests. Second, we will consider a more general scenario where multiple data-rich source cities are available. Finally, we will try to employ RegionTrans in other real-world applications, e.g., facility deployment, where deep spatio-temporal learning is still applicable.

References

[Chen *et al.*, 2016] Longbiao Chen, Daqing Zhang, Leye Wang, Dingqi Yang, Xiaojuan Ma, Shijian Li, Zhaohui Wu, Gang Pan, Thi-Mai-Trang Nguyen, and Jérémie Jakubowicz. Dynamic cluster-based over-demand prediction in bike sharing systems. In *UbiComp*, pages 841–852, 2016.

[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. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 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.

[Li *et al.*, 2015] Yexin Li, Yu Zheng, Huichu Zhang, and Lei Chen. Traffic prediction in a bike-sharing system. In *SIGSPATIAL*, page 33, 2015.

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

[Mathieu *et al.*, 2016] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2016.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 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 *Advances in neural information processing systems*, pages 802–810, 2015.

[Silva *et al.*, 2015] Ricardo Silva, Soong Moon Kang, and Edoardo M Airoldi. Predicting traffic volumes and estimating the effects of shocks in massive transportation systems. *Proceedings of the National Academy of Sciences*, 112(18):5643–5648, 2015.

[Wang *et al.*, 2017a] Leye Wang, Xu Geng, Jintao Ke, Chen Peng, Xiaojuan Ma, Daqing Zhang, and Qiang Yang. Ridesourcing car detection by transfer learning. *arXiv preprint arXiv:1705.08409*, 2017.

[Wang *et al.*, 2017b] 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 Transactions on Intelligent Systems and Technology*, 9(2):20, 2017.

[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 Transactions on Intelligent Systems and Technology*, 7(3):30, 2016.

[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 Transactions on Intelligent Systems and Technology (TIST)*, 5(3):38, 2014.