A Sequential Convolution Network for Population Flow Prediction with Explicitly Correlation Modelling

Jie Feng¹, Ziqian Lin¹, Tong Xia¹, Funing Sun², Diansheng Guo² and Yong Li³
¹Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.
²Tencent Inc., Beijing 100084, China. liyong07@tsinghua.edu.cn

Abstract
Population flow prediction is one of the most fundamental components in many applications from urban management to transportation schedule. It is a challenging task due to the complicated spatial-temporal correlation. While many studies have been done in recent years, they fail to simultaneously and effectively model the spatial correlation and temporal variations among population flows. In this paper, we propose Convolution based Sequential and Cross Network (CSCNet) to solve these difficulties. On the one hand, we design a CNN based sequential structure with progressively merging the flow features from different time in different CNN layers to model the spatial-temporal information simultaneously. On the other hand, we make use of the transition flow as the proxy to efficiently and explicitly capture the dynamic correlation between different types of population flows. Extensive experiments on 4 datasets demonstrate that CSCNet outperforms the state-of-the-art baselines by reducing the prediction error around 7.7%~10.4%.

1 Introduction
Population flow prediction is one of the most fundamental tasks in the urban system and is widely used in many practical applications from urban management, transportation, to resource scheduling in the ride-sharing platform [Yao et al., 2018; Geng et al., 2019]. Being called as crowd flow in urban management, it can be utilized to monitor the anomaly in the group aggregation activities to prevent the accident in time [Zhang et al., 2017]. Being called as traffic prediction in the transportation system, it supports the timely sensing of traffic to meet the travel demand and improve the transportation efficiency [Yao et al., 2019b].

Recently, deep learning has been successfully applied to the modeling of population flow and has become the de facto standard method in many scenarios. As the early application of deep learning in this field, Zhang et al. [Zhang et al., 2016; Zhang et al., 2017] propose to utilize stacked CNNs to model the non-linear spatial dependence by processing the population flow from different time as multi-channel images. Another branch of research mainly relies on the recurrent neural network to model the temporal variation of population flow while using simple conv unit [Yao et al., 2018; Yao et al., 2019a] or attention unit [Qin et al., 2017] as the local spatial feature extractor for a single region. In the third branch, by combining the advantages of both CNN and RNN, researchers [Zonoozi et al., 2018; Wang et al., 2019] utilize ConvLSTM [Shi et al., 2015] to jointly model the spatial-temporal features of city-scale population flow. To better model the dynamic correlations between regions, some researchers try to utilize transition flow [Zhang et al., 2019; Yao et al., 2019b] as the auxiliary feature and achieve the state-of-the-art results in population flow prediction task. Fig. 1 presents the concept of transition flow, which represents the dynamic interaction between regions.

However, while previous deep learning works achieve promising performance, existing methods have at least two limitations. First, existing methods learn the spatial correlation and temporal variations separately thus cannot capture their dependency efficiently. With the movement of population, the spatial and temporal dependency become the most important and challenging characteristics of population flow modelling. While previous works design various models [Zhang et al., 2017; Lin et al., 2019; Yao et al., 2018; Yao et al., 2019a] of CNN and LSTM to capture them, these methods treat them as two kinds of features and build separate units to process them. While ConvLSTM [Zonoozi et al., 2018; Wang et al., 2019] provides a choice to jointly model the spatial-temporal features, its performance is limited [Yao et al., 2019a; Lin et al., 2019]. Second, existing methods ignore the native relation between different types of population flows and fail to model the dynamic correlation between them effectively. Actually, different types of population flows are highly correlated: the out-flow of all regions also make up the in-flow of them, which can be captured by the population transition flow. Existing works ignore this important characteristic and only combine them with simple ways like...
concatenating in the channel dimension [Zhang et al., 2019; Yao et al., 2019b].

In this paper, we design a novel framework, CSCNet, to address the former challenges and achieve better performance for population flow prediction. To simultaneously model the spatial-temporal dependence among population flow of different regions (limitation 1), we propose, Flow Recurrent Network (FRN), to utilize the depth of convolution layers to progressively merging population flow features from different time steps. In FRN, different layer convolution is not only responsible for extracting different level spatial feature but also responsible for capturing the sequential correlation of population flow in different time. Furthermore, we propose Feature Cross Network (FCN) to explicitly fuse different types of population flow including the in/out flow and the transition flow via their innate relation (limitation 2), which will enable the modelling of the accurate and dynamic correlation between them. Besides, to handle the high-dimensional and high-dynamic population transition flow for better feature extraction and fusion, we propose an efficient transition flow compression representation and design a Transition Flow Predictor (TFP) to predict it. In summary, our contributions can be summarized as follows,

- We design FRN with a progressive CNN structure to simultaneously model the spatial-temporal feature of population flows. With the specific design in the fusion unit group, FRN is also able to capture the periodicity (e.g., the daily pattern) in the population flow.
- We design FCN to effectively and explicitly modelling the dynamic correlation between different types of population flows. Besides, with an efficient structure of transition flow data, we also design a predictor TFP to generate its dynamic representation for FCN.
- We conduct extensive experiments on four real-life datasets to demonstrate the effectiveness of our proposed CSCNet on the population flow prediction task. Compared with the state-of-the-art algorithms, CSCNet reduces the RMSE of prediction by 7.72% $\sim$ 10.43%.

2 Preliminaries

Following the widely-used grid-based population flow definition from previous works [Zhang et al., 2017], we define the population flow and its prediction problem as follows.

**Definition 1: Population In/Out Flow**

We define the in-flow and out-flow for a grid region $(h, w)$ at $i^{th}$ time interval in the spatial map as follows:

\[
x_{i}^{h,w,in} = \sum_{S \in \mathcal{P}} \left[ g_{i-1} \notin (h, w) \& g_{i} \in (h, w) \right],
\]

\[
x_{i}^{h,w,out} = \sum_{S \in \mathcal{P}} \left[ g_{i-1} \in (h, w) \& g_{i} \notin (h, w) \right],
\]

where $\mathcal{P}$ represents the collection of trajectories, $S = \{g_{1}, \cdots, g_{t} \}$ is a trajectory in $\mathcal{P}$, and $g_{i}$ is the spatial coordinate. $g_{i} \in (h, w)$ means the trajectory point $g_{i}$ lies within the grid region $(h, w)$, and vice versa.

**Definition 2: Population Transition Flow**

As the extension of population flow, we define the population transition flow map of grid region $(h, w)$ at the $i^{th}$ time interval as follows, for each grid $(m, n)$ in it:

\[
y_{i}^{m,n,in} = \sum_{S \in \mathcal{P}} \left[ g_{i-1} \notin (m, n) \& g_{i} \in (h, w) \right],
\]

\[
y_{i}^{m,n,out} = \sum_{S \in \mathcal{P}} \left[ g_{i-1} \in (h, w) \& g_{i} \in (m, n) \right].
\]

Here, all the parameters are similar to Definition 1. At the $i^{th}$ time interval, the population flow in a $H \times W$ grid spatial map is denoted as $X_{i} = \{X_{i}^{in}, X_{i}^{out}\} : \mathbb{R}^{2 \times H \times W}$. The related population transition flow for each region in the spatial map is concatenated and denoted as $Y_{i} = \{Y_{i}^{in}, Y_{i}^{out}\} : \mathbb{R}^{2 \times H \times W}$, where $R = H \times W$ is the number of regions.

**Population Flow Prediction:**

Given the historical observations $\{X_{i}|i = 1, 2, \cdots, k-1\}$ and $\{Y_{i}|i = 1, 2, \cdots, k-1\}$, predict $X_{k}$.

3 Model Design

Fig. 2 presents the framework of our proposed model-CSCNet. It consists of three components: 1) Flow Recurrent Net (FRN) for population flow (in/out flow) feature extraction and naive prediction; 2) Transition Flow Predictor (TFP) for population transition flow modelling; 3) Feature Cross Net (FCN) for fusing features from different types population flow (in/out/transition flow) and explicitly modelling their dynamic correlation.

![Figure 2: The framework of our proposed CSCNet.](image)

**3.1 Population Flow Modelling via FRN**

**Basic Function Unit**

Three standard CNN based function units used in the model, Local Feature Extraction unit, Feature Fusion unit, and Predictor unit, are presented in Fig. 3. All of them are built upon a 3x3 convolution unit and a following relu function. Local feature extraction unit consists of two 3 $\times$ 3 conv units and a following relu function. With a raw population flow map as input, the local extraction unit extracts spatial features from it and outputs primitive city-scale flow features. Feature fusion unit is introduced to fuse features from different sources with different characteristics to obtain a comprehensive representation of them. It first uses a concate unit to directly merge features and then apply a standard conv unit to fuse features to obtain the new feature. Finally, the Predictor unit with a standard conv is utilized to convert features into the predicted population flow map.
Flow Sequential Net

Based on three standard function units introduced above, we make up the flow sequential net (FSN) as the primitive population flow prediction model. With a sequential population flow map as input, we first utilize a shared local extraction unit to extract spatial features of each time step and then stack feature fusion units to sequentially process the spatial features to obtain the global feature map. Finally, the global feature map is converted into the population flow map prediction by a predictor unit.

Different from previous works [Zhang et al., 2017; Lin et al., 2019] who directly regard sequential population flow map as different channels of an image and model them as a whole, we progressively model the sequential relationships between different population flow maps via the shared local extraction unit and sequential fusion units. From another view, our sequential fusion nets can also be regarded as a deeper feature extractor which is helpful for the final prediction. Except for the predictor unit, the feature map generated by the final fusion unit can also be directly output to other components like feature cross net for further processing.

Flow Recurrent Net

In Fig. 4, we present the design of FRN as the advanced recurrent version of FSN. Many previous works [Zhang et al., 2017; Zonoozi et al., 2018; Yao et al., 2019a] have demonstrated that periodicity is one of the most distinguishing characteristics in the population flow prediction problem and the most important periodicity is the daily pattern. Based on this observation, we upgrade FSN as FRN to capture the daily pattern of population flow.

In FRN, we introduce a periodical fusion unit group to force each fusion unit to focus on modeling the population flow feature of the specific time step. All the periodical fusion unit in the group work together to cover the whole periodicity of population flow. When feeding population flow from different time step, the related fusion unit with the specific period pattern is triggered to fuse the local features. Fig. 4 presents a simple example, where different population flow from different days with the same hour of a day will use the same fusion unit for feature fusing. It is noted that the normal input data of FRN is the population in/out flow and we can also extend it with the transition flow in the channel dimension (denoted as CSCNet-RTC+ in the evaluation section).

3.2 Transition Flow Modelling via TFP

Truncated Transition Flow

As Fig. 5 shows, we arrange the transition flow map of $H \times W$ grid regions in the $z$-axis direction to obtain the original transition flow tensor. While it is similar to [Zhang et al., 2019], we introduce two different designs for effective feature extraction. First, without combining the in/out transition flow map in one tensor, we construct two separate transition map tensors and leave their fusion in the following network.

More importantly, we design the truncated transition flow map to represent the delicate direction information of the population transition flow. While containing more information about the flow, the complete transition flow map is high-dimensional and high-dynamic. Thus, we propose a truncated transition flow map to distill knowledge from it with only preserving local direction information. As Fig. 5 shows, we summarize the direction information in the circle of $r=1$, e.g., the top-right direction contains about 50% of the outflow from the center region. We design two aggregation methods to obtain this normalized truncated transition flow map. The first method accumulates the population variation in each direction, e.g., when we only use the circle of $r=1$ to represent the flow direction, all the population variations outside the circle are accumulated into the nearby regions in the circle. And then we normalize the flow in different directions by the total flow from all directions. The second method omits the population variation outside the circle and only calculates the population variation in the circle as the approximation of flow in different directions. Due to the continuity of mobility, the second method obtains competitive results and computes efficiently in practice.

Transition Flow Predictor

Following the definition of the truncated transition flow map, we design a transition flow predictor (TFP) to generate the
transition flow map on the target time to enable the modeling of dynamic spatial correlation. Based on the basic functional units introduced before, we first utilize a fusion unit to combine the recent \( L \) truncated transition flow map in the channel dimension. Here, different from the former standard fusion unit, we extend the input fusion unit from 2 to \( L \), which is the length of the historical population flow. Then, we stack \( k \) local extraction units to model transition flow features. Finally, we utilize the standard predictor unit to obtain the transition flow prediction at the target time step with different channels denoting the population flow from different directions.

### 3.3 Flow Feature Fusing via FCN

Finally, we introduce how to effectively model the dynamic correlation between different types of population flows (including in/out/transition flow) with explicitly motivation.

**Figure 6**: Feature cross network (FCN) for the inflow branch.

By taking the *inflow feature cross net* branch as an example, the core idea of FCN is that the **inflow feature of a region can be explained as the weighted sum of the outflow features from its nearby regions, which is recorded by the transition flow**. The whole design of the **inflow cross net** is presented in Fig. 6. In FCN, we process the auxiliary outflow feature with transition flow feature to obtain the tuned feature map which can be regarded as a kind of “new” inflow feature, and details of this operation are introduced later. Here, the transition flow features are the dynamic prediction results from TFP. Then, we combine this “new” inflow feature with the original inflow feature to obtain the fused feature for final inflow prediction. To do this, we use two independent conv units to process the original inflow feature and the “new” inflow feature. Then we fuse them via the element-wise addition as the final fused feature.

Fig. 7 presents a simple case showing how to generate the “new” inflow feature from the original outflow feature map by utilizing the transition flow map as the guidance. In the outflow feature map, we take region \( A, B, C \) as examples to calculate the “new” inflow feature map. Based on the inflow transition map of region \( B \), we know that 50% of inflow from region \( B \) is from its bottom direction region \( A \). This indicates that the flow from region \( A \) is more important for region \( B \) than flow from other regions. Thus, we need to assign a bigger weight for region \( A \) in the generation of the inflow of region \( B \). Basically, we use the normalized value in the transition flow map as the initial weight for this goal. The case for region \( C \) presents the outflow from region \( A \) is not so important for the inflow of it. By repeating this operation around the whole outflow map, we can obtain the “new” inflow feature map.

**Figure 7**: Intuitive illustration of applying the transition flow feature to the outflow feature to obtain the “new” inflow feature.

**Outflow Feature Map**

**Region B Inflow Transition Map**

**Region C Transition Map**

**Tuned Feature Map (New Inflow Feature)**

\[ f^\text{new}_{\text{in}}[i; h, w] = \sum_{i=h-r:j=w-r} f^\text{out}[i; j] \circ f^\text{pre}_{\text{trans}}[i; j], h, w, \]

where \( f \) denotes flow feature.

### 3.4 Training

While **CSCNet** contains three independent components for different goals, we train them one by one to obtain the complete model. First, we train two independent FRNs for the population in/out flow prediction. Then, we process the truncated transition flow maps and train two TFPs to predict the future transition flow. Finally, based on the in/out flow feature from FRNs and predicted transition flow from TFPs, we train FCNs for feature fusion and obtain the final results.

### 4 Performance Evaluation

#### 4.1 Datasets

We evaluate our model on four population flow data. The first two datasets MobileBJ and BikeNYC(agg) are from previous work [Lin et al., 2019]. Since these two datasets only contain aggregated population flow data, we construct new population flow data with transition flow from two public data sources: NYCTaxi\(^1\) and NYC\(\text{BIke}\)\(^2\). We extract one-month population data from the public source and follow [Zhang et al., 2017; Lin et al., 2019] to process it. The first three weeks’ data is used for training, and the left is used for testing.

#### 4.2 Baselines

We compare our model with 8 state-of-the-art baselines. The first group with 6 baselines does not consider transition flow: 1) **HA**: Historical Average, which predicts population flow by the average value of historical flow; 2) **ARIMA** [Box et al., 2015]: One of the most classic methods for time series modeling; 3) **ConvLSTM** [Shi et al., 2015]: It combines the convolution and LSTM to capture both the spatial and temporal features simultaneously; 4) **Perordic-CRN** [Zonoozi et al., 2018]: It stacks several ConvGRU layers with the pyramidal structure and also consider the memory based periodic representation; 5) **ST-ResNet** [Zhang et al., 2017]: It

\(^1\) https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

\(^2\) https://www.citibikenyc.com/system-data
uses stacked CNNs with residual connection to model population flows as images with different channels; 6) Deep-STN+ [Lin et al., 2019]: It extends ST-ResNet by modeling long-range dependence and semantic effects. The second group with 2 baselines utilizes the transition flow information: 1) STDN [Yao et al., 2019b]: With attention for modeling periodicity shift, it also designs a flow gating function to fuse the transition flow feature for traffic prediction; 2) MDL [Zhang et al., 2019]: With the similar backbone of ST-ResNet, it utilizes the multi-task learning framework to predict in/out flow and transition flow simultaneously. Five variants of proposed CSCNet are 1) CSCNet-S: the most basic version of CSCNet, which only contains a FSN; 2) CSCNet-R: the advanced version of CSCNet-S, which replaces FSN with a FRN considering the periodicity; 3) CSCNet-RC: the combination of FRN and FCN, where TFP is removed and historical transition flow is directly fed to FCN; 4) CSCNet-RTC: which contains all the special designed components: FRN, TFP and FCN; 5) CSCNet-RTC+: the transition flow data is also fed into FRN by directly concatenating in the channel dimension.

**Metrics and Parameter Settings**

We use Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) as metrics, which are as follows:

\[
RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} ||X_i - \hat{X}_i||_2}, \quad MAE = \frac{1}{N} \sum_{i=1}^{N} |X_i - \hat{X}_i|,
\]

where \(X_i\) and \(\hat{X}_i\) denote the ground-truth and the prediction at the \(i^{th}\) time interval. \(N\) is the total number of samples in the testing data. RMSE is also used as the loss function.

The basic parameters of conv can refer to Fig. 3 and the default output channel of each conv is 64. In FRN, the default number of fusion unit is 24 and the default length of sequential input is 24. In FCN, the default spatial range of transition flow is \(r = 1\). For different baselines, we follow the suggested parameters int the original papers. Besides, all the reported results in the experiments are the average of at least 5 independent runnings.

### 4.3 Experiment Results

**Performance of FSN&FRN** We first compare CSCNet-S and CSCNet-R with baselines on the MobileBJ and BikeNYC data to present the effectiveness of FSN and FRN in Table 1. By simultaneously modeling the spatial-temporal variations with progressively fusing the population flow from different time intervals in different layers, CSCNet-S outperforms the best baseline DeepSTN+ by reducing the RMSE 1.8% \(\sim 7.8\%). Considering the periodicity in the model design, CSCNet-R outperforms CSCNet-S and further reduces the prediction error. Besides, compared with the ConvSTM based models like Perodic-CRN, which is another choice for simultaneously spatial-temporal modeling, our model CSCNet-R with progressively CNN structure also performs much better. Based on the results of two datasets in the top 7 rows in Table 2, we observe that our proposed FRN (CSCNet-R) continuously outperforms than all the baselines with more than 3% prediction error reduction.

#### Table 1: Comparison of different baselines and variants of CSCNet on new NYCTaxi and NYCBIke data.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>RMSE</th>
<th>Δ RMSE</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BikeNYC (agg)</td>
<td>ARIMA</td>
<td>10.89</td>
<td>82.94%</td>
<td>3.25</td>
</tr>
<tr>
<td></td>
<td>ConvLSTM</td>
<td>6.41</td>
<td>74.67%</td>
<td>2.54</td>
</tr>
<tr>
<td></td>
<td>Perodic-CRN</td>
<td>6.37</td>
<td>63.88%</td>
<td>2.70</td>
</tr>
<tr>
<td></td>
<td>ST-ResNet</td>
<td>6.48</td>
<td>84.73%</td>
<td>2.40</td>
</tr>
<tr>
<td></td>
<td>DeepSTN+</td>
<td>5.96</td>
<td>0.00%</td>
<td>2.29</td>
</tr>
<tr>
<td>ARIMA</td>
<td>5.85</td>
<td>-1.84%</td>
<td>2.33</td>
<td></td>
</tr>
<tr>
<td>ConvLSTM</td>
<td>5.73</td>
<td>-3.86%</td>
<td>2.31</td>
<td></td>
</tr>
<tr>
<td>BikeNYC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CSCNet-S</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CSCNet-R</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARIMA</td>
<td>58.63</td>
<td>58.93%</td>
<td>30.05</td>
<td></td>
</tr>
<tr>
<td>ConvLSTM</td>
<td>44.31</td>
<td>44.43%</td>
<td>27.75</td>
<td></td>
</tr>
<tr>
<td>Perodic-CRN</td>
<td>41.22</td>
<td>44.43%</td>
<td>27.88</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ST-ResNet</td>
<td>42.19</td>
<td>14.37%</td>
<td>26.95</td>
</tr>
<tr>
<td></td>
<td>DeepSTN+</td>
<td>36.89</td>
<td>0.00%</td>
<td>23.43</td>
</tr>
<tr>
<td>CSCNet-S</td>
<td>34.00</td>
<td>-7.83%</td>
<td>21.05</td>
<td></td>
</tr>
<tr>
<td>CSCNet-R</td>
<td>33.25</td>
<td>-9.87%</td>
<td>21.71</td>
<td></td>
</tr>
</tbody>
</table>

**Performance of TFP&FCN** We compare CSCNet with more state-of-the-art baselines on NYCBIke and NYCTaxi data in Table 2 to evaluate the effectiveness of TFP and FCN. First, we can observe that STDN and MDL with utilizing transition flow data achieve better results than the previous best baseline DeepSTN+, which presents similar performance as CSCNet-R. Second, CSCNet-RTC performs much better than the best baseline MDL by reducing 6.2%/3.1% prediction error and also better than STDN by reducing 10.98%/7.22% prediction error on NYCBIke/NYCTaxi data. The superior performance of CSCNet-RTC demonstrates the effectiveness of our design in explicitly modelling the dynamic correlation between different types of population flows (in/out/transition flow). Third, CSCNet-RTC without TFP performs worse than the CSCNet-RTC, which presents the necessity of constructing and predicting the dynamic transition flow for efficient feature extraction. Finally, based on CSCNet-RTC, CSCNet-RTC+ uses the transition flow as the additional input of FRN and improves the performance again.

**Hyper-parameter Study** We conduct a hyper-parameter study of several key components in our model on NYCBIke data to evaluate the effectiveness of them. The results on NYCTaxi data is similar, and we omit it here due to the

#### Table 2: Comparison of different baselines and variants of CSCNet on new NYCTaxi and NYCBIke data.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>NYCBIke</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>RMSE</td>
<td>Δ RMSE</td>
<td>MAE</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>HA</td>
<td>15.11</td>
<td>34.67%</td>
<td>49.62</td>
</tr>
<tr>
<td>ARIMA</td>
<td>14.56</td>
<td>29.77%</td>
<td>44.43</td>
</tr>
<tr>
<td>ConvLSTM</td>
<td>11.57</td>
<td>3.12%</td>
<td>44.39</td>
</tr>
<tr>
<td>ST-ResNet</td>
<td>11.60</td>
<td>3.39%</td>
<td>37.52</td>
</tr>
<tr>
<td>DeepSTN+</td>
<td>11.22</td>
<td>0.00%</td>
<td>36.90</td>
</tr>
<tr>
<td>CSCNet-S</td>
<td>11.28</td>
<td>0.53%</td>
<td>37.30</td>
</tr>
<tr>
<td>CSCNet-R</td>
<td>10.84</td>
<td>-3.39%</td>
<td>36.19</td>
</tr>
<tr>
<td>RA-STDN</td>
<td>10.29</td>
<td>0.61%</td>
<td>36.70</td>
</tr>
<tr>
<td>RA-MDL</td>
<td>10.85</td>
<td>-3.30%</td>
<td>35.89</td>
</tr>
<tr>
<td>RA-CSCNet-RC</td>
<td>10.31</td>
<td>-8.11%</td>
<td>35.46</td>
</tr>
<tr>
<td>RA-CSCNet-RTC</td>
<td>10.17</td>
<td>-9.36%</td>
<td>34.75</td>
</tr>
<tr>
<td>RA-CSCNet-RTC+</td>
<td>10.05</td>
<td>-10.43%</td>
<td>34.05</td>
</tr>
</tbody>
</table>
space limitation. Fig. 8(a) presents the effects of the length of the population flow, we observe that the performance of the model is improved rapidly with the increase of length, where the length reaches 8 is enough for competitive performance. As Fig. 8(b) shows, only 1 conv in the fusion unit is good enough to obtain the best performance. Fig. 8(c) presents the effects of the number of fusion units in the periodical FRN. If we only use a small number of fusion units (e.g., 6) by repeating them to construct the 24 fusion units in FRN, the performance of our model is limited obviously, which represents the importance of daily periodicity in population flow modelling. Finally, we evaluate the effects of the spatial range of truncated transition flow in FCN in Fig. 8(d), where spatial range \( r = 0 \) means no transition flow data is used and spatial range \( r = 1 \) means that we only consider the transition flow in the eight directions. Based on Fig. 8(d), we observe the utilization of transition flow can significantly reduce the prediction error. Besides, we also notice that increase the spatial range does not improve the performance again.

(c) Effects of the number of reused fusion units in FRN. (d) Effects of the spatial range of transition flow in FCN.

Figure 8: Parameter study of CSCNet on NYCBike data.

5 Related Work

Various conventional algorithms [Fan et al., 2015; Xu et al., 2016b; Xu et al., 2016a; Hoang et al., 2016; Xia et al., 2020] were proposed to model the temporal variation with considering simple spatial correlation. With the success of deep learning techniques [Wei et al., 2018; Feng et al., 2018b; Feng et al., 2018a], it was applied to the population flow prediction and became the de facto standard method in recent three years. DeepST [Zhang et al., 2016], ST-ResNet [Zhang et al., 2017] proposed to utilize CNN to model the spatial correlation between different regions in the city scale as images. Following their framework, DeepSTN+ [Lin et al., 2019] proposed to model the long-range dependencies and the semantic effects of region function. DMVST [Yao et al., 2018] and Meta-ST [Yao et al., 2019a] utilized shallow CNN as the local spatial feature extractor for the target region and then made use of LSTM to model the temporal variations of it. Furthermore, RegionTrans [Wang et al., 2019] and Periodic-CRN [Zonoozi et al., 2018] proposed to utilize ConvLSTM [Shi et al., 2015] and its variants to model the spatial-temporal feature simultaneously. Different from ConvLSTM, which extends LSTM by integrating convolutional operation to it for capturing both spatial and temporal information, we propose a new paradigm of simultaneously spatial-temporal modelling by only using the convolutional structure with progressively merging features from different time in different convolution layer. In this way, going deeper with convolution [Szegedy et al., 2015] in our model is not only for better spatial feature extraction but also for progressively merging the temporal feature.

Recently, researchers [Zhang et al., 2019; Yao et al., 2019b; Yu et al., 2018; Li et al., 2017; Geng et al., 2019] proposed to utilize auxiliary information to model the complicated correlation between different regions. Some [Yu et al., 2018; Li et al., 2017; Geng et al., 2019] constructed this correlation by applying graph convolution neural network on the static road network. Furthermore, MDL [Zhang et al., 2019] and STDN [Yao et al., 2019b] were proposed to make use of the dynamic population transition flow data to model the flexible and dynamic spatial correlations and achieves the state-of-the-art performance. Compared with existing works, we propose CNN based sequential network to simultaneously model the spatial-temporal features, which enhance the sequential modelling capacity of CNN while preserving the advantages of global spatial modeling and parallel computing. Further, we propose to predict the simplified transition flow feature and design a feature cross network for explicitly reorganizing the original flow feature to capture the dynamic correlation between regions.

6 Conclusion

In this paper, we investigate the population flow prediction problem. We propose CSCNet with FRN to simultaneously model the spatial-temporal feature and FCN to explicitly capture the correlations between different types of population flows. We evaluate our model on four real-life datasets, which shows that our model outperforms all the state-of-the-art baselines significantly. In the future, we will consider more flexible geometric models like graph neural networks to extend the application scenario of the proposed methods.

Acknowledgements

This work was supported in part by The National Key Research and Development Program of China under grant 2018YFB1800804, the National Nature Science Foundation of China under U1936217, 61971267, 61972223, 61941117, 6186136003, Beijing Natural Science Foundation under L182038, Beijing National Research Center for Information Science and Technology under 20031887521, and research fund of Tsinghua University - Tencent Joint Laboratory for Internet Innovation Technology.
References


[Xia et al., 2020] Tong Xia, Yong Li, Yue Yu, Fengli Xu, Qingmin Liao, and Depeng Jin. Understanding urban dynamics via state-sharing hidden markov model. IEEE Transactions on Knowledge and Data Engineering, 2020.


