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

Jie Feng[1] , **Ziqian Lin**[1] , **Tong Xia**[1] , **Funing Sun**[2] , **Diansheng Guo**[2] , **Yong Li**[1]
[1]Beijing National Research Center for Information Science and Technology (BNRist),
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China.
[2]Tencent 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 **C**onvolution based **S**equential and **C**ross 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]. With the popularity of the ride-sharing platform, it also becomes an essential ability for the supply-demand based intelligent resource schedule mechanism [Wang *et al.*, 2017; Yao *et al.*, 2018]. Due to the great value in reality, many prediction methods have been proposed in the last decades. Before the prevalent deep learning techniques, researchers focused on traditional time series modeling [Li *et al.*, 2012; Xu *et al.*, 2016b] and simple local spatial dependence [Hoang *et al.*, 2016; Deng *et al.*, 2016], which failed to model the non-linear spatial-temporal dependency between different population flows.

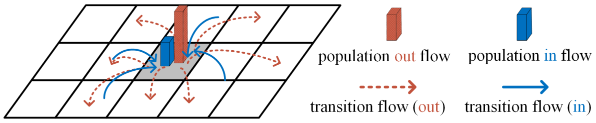


Figure 1: Illustration of the population in/out/transition flow.

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; Feng *et al.*, 2018a] 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, **F**low **R**ecurrent **N**etwork (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 **F**eature **C**ross **N**etwork (FCN) to explicitly fuse different types of population flow including the in/out flow and the transition flow via their native 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 **T**ransition **F**low **P**redictor (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; Yao *et al.*, 2018; Lin *et al.*, 2019], 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:

$$x_i^{h,w,in} = \sum\nolimits_{S\in\mathbb{P}} |g_{i-1} \notin (h,w) \;\;\&\;\; g_i \in (h,w)|,$$
$$x_i^{h,w,out} = \sum\nolimits_{S\in\mathbb{P}} |g_{i-1} \in (h,w) \;\;\&\;\; g_i \notin (h,w)|,$$

where $\mathbb{P}$ represents the collection of trajectories, $S = \{g_1, \cdots, g_i, \cdots, g_{|S|}\}$ is a trajectory in $\mathbb{P}$, 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:

$$y_{h,w,i}^{m,n,in} = \sum\nolimits_{S\in\mathbb{P}} |g_{i-1} \in (m,n) \;\;\&\;\; g_i \in (h,w)|,$$
$$y_{h,w,i}^{m,n,out} = \sum\nolimits_{S\in\mathbb{P}} |g_{i-1} \in (h,w) \;\;\&\;\; g_i \in (m,n)|.$$

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 $\mathbf{X}_i = (\mathbf{X}_i^{in}, \mathbf{X}_i^{out}) : \mathbb{R}^{2\times H\times W}$. The related population transition flow for each region in the spatial map is concatenated and denoted as $\mathbf{Y}_i = (\mathbf{Y}_i^{in}, \mathbf{Y}_i^{out}) : \mathbb{R}^{2\times R\times H\times W}$, where $R = H \times W$ is the number of regions.

Population Flow Prediction:

Given the historical observations $\{\mathbf{X}_i | i = 1, 2, \cdots, k-1\}$ and $\{\mathbf{Y}_i | i = 1, 2, \cdots, k-1\}$, predict $\mathbf{X}_k$.

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.

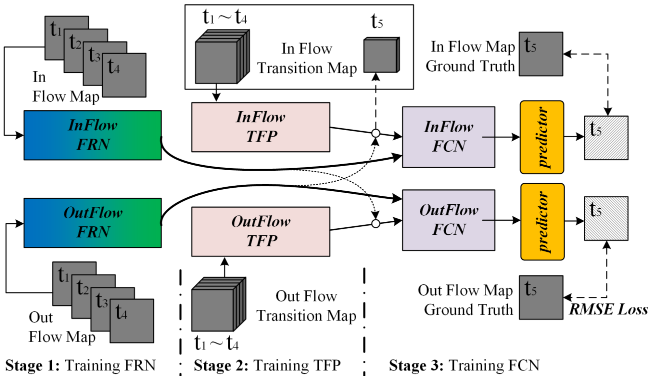


Figure 2: The framework of our proposed *CSCNet*.

3.1 Population Flow Modelling via FRN

In this section, we first introduce several basic functional units of *CSCNet* and then utilize them to make up the flow recurrent net (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. The basic parameters of 3x3 convolution unit include $kernel\ size = 3$, $stride = 1$, and $padding = 1$. Here, we do not use any pooling function, thus preserving the feature size with the original size of $H \times W$. Based on this standard $conv$ unit, we build three basic function units for extraction, fusion and prediction as follows. *Local feature extraction unit* consists of two 3×3 $conv$ 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.

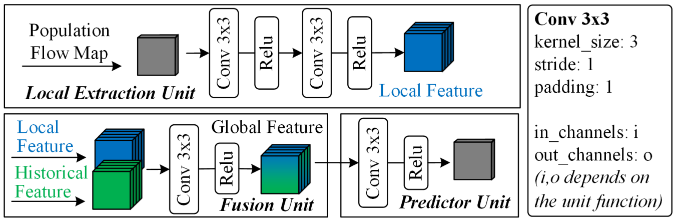


Figure 3: Three basic units of *CSCNet*, where "local" refers to a short-term time step and "global" refers to a long-term time window with several time steps.

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.

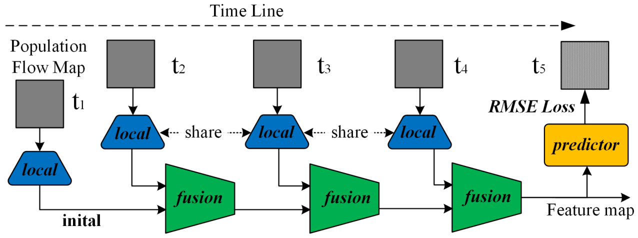


Figure 4: Basic sequential modeling framework (FSN) of *CSCNet*.

Flow Recurrent Net

In Fig. 5, 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.

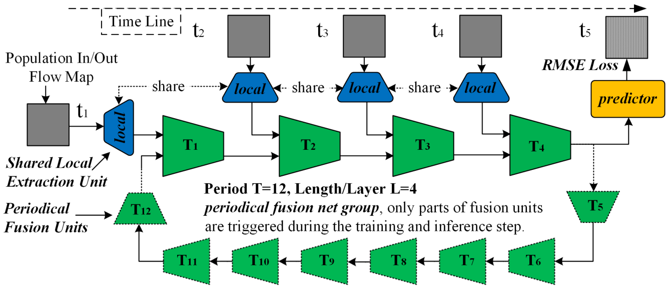


Figure 5: Flow recurrent net (FRN) in *CSCNet*.

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. 5 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. In Fig. 5, we build 12 fusion nets to model the different relationships between population flows in different time periods (2 hours as one time step). With a 4 slices population flow ($t_1 \sim t_4$) as input, local spatial features are first extracted from the shared local feature extraction unit. Then, four local features ($t_1 \sim t_4$) are processed by four specific fusion units ($T_1 \sim T_4$) with the same time labels. Finally, the fused feature map is fed into the predictor unit to obtain the prediction result. When we input the population flow from $t_2 \sim t_5$, the activated fusion units become ($T_2 \sim T_5$) and the local feature extractor and the predictor unit keeps unchanged. 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

In this section, we introduce how to process the population transition flow and design a simple transition flow predictor to predict transition flow in the next time step for final fusion.

Truncated Transition Flow

As mentioned before, the population transition flow is high-dimensional and sparse. High-dimension of population flow path means that for a city with $H \times W$ grids, the size of population transition flow map is up to $H \times W \times H \times W$. As the left flow map in Fig. 6 shows, each grid region in the city will have a $H \times W$ transition (in) flow map to represent the details of the population (in) flow. Furthermore, as Fig. 6 shows, the transition flow map for each region is always very sparse which means only limited regions have values. Both of these make the processing of transition flow challenging. Based on the statistics of transition flow on the real world mobility data, we observe that the range of influence of the transition flow for each region is limited to its adjacent regions, e.g., more than 60% flow is from/to the adjacent regions within 1-hot distance. As Fig. 6 shows, we first 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.

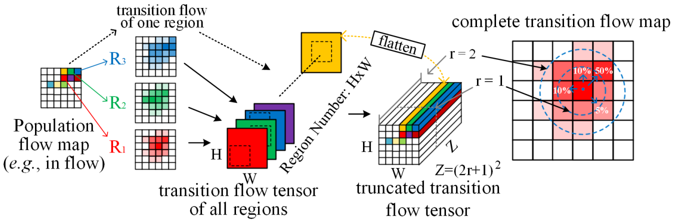


Figure 6: Truncated transition flow map construction.

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. With this normalization operation, the value in the truncated flow path map is limited into $[0, 1]$, which benefits for the following processing and model training. As Fig. 6 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.

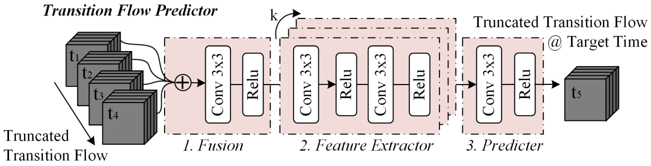


Figure 7: The flow path predictor for *CSCNet*.

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. The structure of TFP is presented in Fig. 7.

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.

Existing works [Zhang *et al.*, 2017; Zonoozi *et al.*, 2018; Yao *et al.*, 2019a] on population flow map prediction combine the inflow and outflow in the different channel of a unified flow map tensor and utilize various CNN models to directly capture the variation of these flows. This operation omits the organic correlation between flows and fails to obtain better performance. Different from them, we propose to utilize the transition flow map to explicitly capture the correlation between the inflow and outflow to better model the population mobility pattern and obtain more accurate prediction results.

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. 8. 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

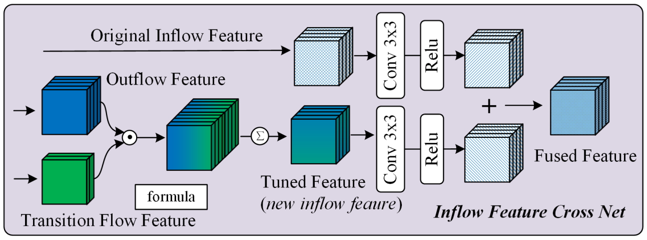


Figure 8: Feature cross network (FCN) for the inflow branch.

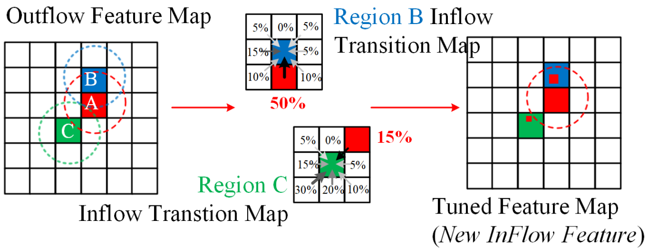


Figure 9: Intuitive illustration of applying the transition flow feature to the outflow feature to obtain the "new" inflow feature.

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. 9 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 of 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. For region (h, w) in the "new" inflow feature map, its value is calculated as follows,

$$\mathbf{flow}^{\mathbf{new}}_{\mathbf{in}}[:,h,w] = \sum_{i=h-r,j=w-r}^{h+r,w+r} \mathbf{flow}_{\mathbf{out}}[:,i,j] \circ \mathbf{flow}^{\mathbf{pre}}_{\mathbf{trans}}[id_{(i,j)},h,w],$$

$$id_{(i,j)} = (i-(h-r))*(2r+1)+(j-(w-r)).$$

where $flow_{in}^{new}$ is the new in-flow feature, $flow_{out}$ denotes the outflow feature from FRN, $flow_{trans}^{pre}$ is the transition flow feature from TFP, and r denotes the spatial range of transition flow feature.

3.4 Training

While *CSCNet* contains three independent components for different goals, we train them one by one to obtain the final 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 prediction 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 NYCBike[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 6 baselines only consider the population in/out flow.

- **HA**: Historical Average, which predicts population flow by the average value of historical flow.
- **ARIMA** [Box *et al.*, 2015]: One of the most classic methods for time series modeling.
- **ConvLSTM** [Shi *et al.*, 2015]: It combines the convolution and LSTM to capture both the spatial and temporal features simultaneously.
- **Peoridic-CRN** [Zonoozi *et al.*, 2018]: It stacks several ConvGRU layers with the pyramidical structure and also consider the memory based periodic representation.
- **ST-ResNet** [Zhang *et al.*, 2017]: It uses stacked CNNs with residual connection to model population flows as images with different channels.
- **DeepSTN+** [Lin *et al.*, 2019]: It extends ST-ResNet by modeling long-range dependence and semantic effects.

The left 2 baselines utilize the transition flow information to improve performance.

- **STDN** [Yao *et al.*, 2019b]: With attention for modelling periodicity shift, it also designs a flow gating function to fuse the transition flow feature for traffic prediction.
- **MDL** [Zhang *et al.*, 2019]: With the similar backbone of ST-ResNet, it utilizes the multi-task learning to predict the population flow and the transition flow simultaneously.

Five variants of proposed *CSCNet* are as follows.

- ***CSCNet-S***: most basic version of *CSCNet*, which only contains a **FSN** (Flow Sequential Network).
- ***CSCNet-R***: advanced version of *CSCNet-S*, which replaces FSN with a **FRN** considering the periodicity.
- ***CSCNet-RC***: combination of **FRN** and **FCN**, where TFP is removed and transition flow is directly fed to FCN.
- ***CSCNet-RTC***: which contains all the special designed components: **FRN**, **TFP** and **FCN**.

[1] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
[2] https://www.citibikenyc.com/system-data

- ***CSCNet-RTC+***: the transition flow data is also fed into FRN by directly concating 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:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}||\mathbf{X}_i - \widehat{\mathbf{X}}_i||_2^2}, MAE = \frac{1}{N}\sum_{i=1}^{N}|\mathbf{X}_i - \widehat{\mathbf{X}}_i|,$$

where $\mathbf{X}_i$ and $\widehat{\mathbf{X}}_i$ 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. The results of baselines are from the original paper [Lin *et al.*, 2019] and the performance of DeepSTN+ in the tale includes the semantic parts while we do not consider the semantic effects of PoIs in our model. 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 ConvLSTM based models like Peoridic-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.

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 $6.27\%/3.18\%$ 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-RC* 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.

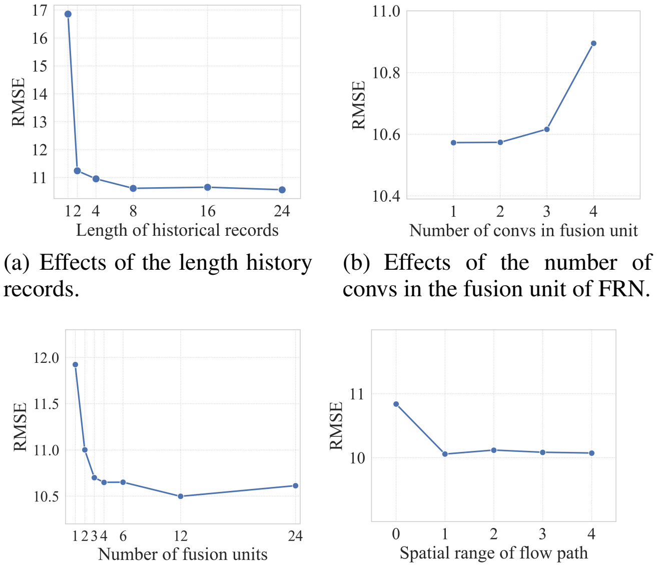
Table 1: Comparison of different baselines and variants of *CSCNet* on MobileBJ and BikeNYC(agg).

Dataset	Model	RMSE	Δ RMSE	MAE
BikeNYC (agg)	ARIMA	10.89	82.94%	3.25
	ConvLSTM	6.41	7.67%	2.54
	Peoridic-CRN	6.37	6.88%	2.70
	ST-ResNet	6.48	8.73%	2.40
	DeepSTN+	5.96	0.00%	**2.29**
	CSCNet-S	5.85	-1.84%	2.33
	CSCNet-R	**5.73**	**-3.86%**	**2.31**
MobileBJ	ARIMA	58.63	58.93%	30.05
	ConvLSTM	44.31	20.11%	27.75
	Peoridic-CRN	41.22	11.74%	27.88
	ST-ResNet	42.19	14.37%	26.95
	DeepSTN+	36.89	0.00%	23.43
	CSCNet-S	34.00	-7.83%	**21.05**
	CSCNet-R	**33.25**	**-9.87%**	**21.71**

Table 2: Comparison of different baselines and variants of *CSCNet* on new NYCTaxi and NYCBike data.

	Datasets	**NYCBike**		**NYCTaxi**	
	Model	**RMSE**	**Δ RMSE**	**RMSE**	**Δ RMSE**
w/o transition flow	HA	15.11	34.67%	49.62	34.47%
	ARIMA	14.56	29.77%	44.43	20.41%
	ConvLSTM	11.57	3.12%	44.39	20.30%
	ST-ResNet	11.60	3.39%	37.52	1.68%
	DeepSTN+	11.22	0.00%	36.90	0.00%
	CSCNet-S	11.28	0.53%	37.30	1.08%
	CSCNet-R	10.84	-3.39%	36.19	-1.92%
with transition flow	STDN	11.29	0.61%	36.70	-0.54%
	MDL	10.85	-3.30%	35.89	-2.74%
	CSCNet-RC	10.31	-8.11%	35.46	-3.90%
	CSCNet-RTC	10.17	-9.36%	34.75	-5.83%
	CSCNet-RTC+	**10.05**	**-10.43%**	**34.05**	**-7.72%**

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 space limitation. Fig. 10(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. 10(b) shows, only 1 conv in the fusion unit is good enough to obtain the best performance. Fig. 10(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. 10(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. 10(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.



(a) Effects of the length history records.

(b) Effects of the number of convs in the fusion unit of FRN.

(c) Effects of the number of reused fusion units in FRN.

(d) Effects of the spatial range of transition flow in FCN.

Figure 10: 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. DeepSD [Wang *et al.*, 2017] tried to combine the techniques of external factor embedding, multiple linear layers with residual connection to predict the order for the region. 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. While previous works succeeded in modeling the spatial correlation from the city scale, they only construct manual temporal feature in the input. 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 simplify and predict the transition flow with minimal direction information and design feature cross network by explicitly using this information to reorganize 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.

References

[Box *et al.*, 2015] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[Deng *et al.*, 2016] Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, Linhong Zhu, Rose Yu, and Yan Liu. Latent space model for road networks to predict time-varying traffic. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1525–1534. ACM, 2016.

[Fan *et al.*, 2015] Zipei Fan, Xuan Song, Ryosuke Shibasaki, and Ryutaro Adachi. Citymomentum: an online approach for crowd behavior prediction at a citywide level. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2015.

[Feng *et al.*, 2018a] Jie Feng, Xinlei Chen, Rundong Gao, Ming Zeng, and Yong Li. Deeptp: An end-to-end neural network for mobile cellular traffic prediction. *IEEE Network*, 32(6):108–115, 2018.

[Feng *et al.*, 2018b] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*, pages 1459–1468, 2018.

[Geng *et al.*, 2019] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *2019 AAAI Conference on Artificial Intelligence (AAAI'19)*, 2019.

[Hoang *et al.*, 2016] Minh X Hoang, Yu Zheng, and Ambuj K Singh. Forecasting citywide crowd flows based on big data. *ACM SIGSPATIAL*, 2016.

[Li *et al.*, 2012] Xiaolong Li, Gang Pan, Zhaohui Wu, Guande Qi, Shijian Li, Daqing Zhang, Wangsheng Zhang, and Zonghui Wang. Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science*, 6(1):111–121, 2012.

[Li *et al.*, 2017] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

[Lin *et al.*, 2019] Ziqian Lin, Jie Feng, Ziyang Lu, Yong Li, and Depeng Jin. Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1020–1027, Jul. 2019.

[Qin *et al.*, 2017] Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang, and Garrison W Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2627–2633. AAAI Press, 2017.

[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.

[Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[Wang *et al.*, 2017] Dong Wang, Wei Cao, Jian Li, and Jieping Ye. Deepsd: supply-demand prediction for online car-hailing services using deep neural networks. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 243–254. IEEE, 2017.

[Wang *et al.*, 2019] Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. Cross-city transfer learning for deep spatio-temporal prediction. In *IJCAI*, 2019.

[Wei *et al.*, 2018] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

[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.

[Xu *et al.*, 2016a] Fengli Xu, Jie Feng, Pengyu Zhang, and Yong Li. Context-aware real-time population estimation for metropolis. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 1064–1075. ACM, 2016.

[Xu *et al.*, 2016b] Fengli Xu, Yuyun Lin, Jiaxin Huang, Di Wu, Hongzhi Shi, Jeungeun Song, and Yong Li. Big data driven mobile traffic understanding and forecasting: A time series approach. *IEEE transactions on services computing*, 9(5):796–805, 2016.

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

[Yao *et al.*, 2019a] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *The World Wide Web Conference*, 2019.

[Yao *et al.*, 2019b] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *AAAI*, volume 33, pages 5668–5675, 2019.

[Yu *et al.*, 2018] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3634–3640. AAAI Press, 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 *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 92. ACM, 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.

[Zhang *et al.*, 2019] Junbo Zhang, Yu Zheng, Junkai Sun, and Dekang Qi. Flow prediction in spatio-temporal networks based on multitask deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[Zonoozi *et al.*, 2018] Ali Zonoozi, Jung-jae Kim, Xiao-Li Li, and Gao Cong. Periodic-crn: A convolutional recurrent model for crowd density prediction with recurring periodic patterns. In *IJCAI*, pages 3732–3738, 2018.