

# Attentive Crowd Flow Machines

Lingbo Liu  
Sun Yat-sen University  
liulingb@mail2.sysu.edu.cn

Guanbin Li\*  
Sun Yat-sen University  
liguanbin@mail.sysu.edu.cn

Ruimao Zhang  
The Chinese University of Hong Kong  
ruimao.zhang@iee.org

Bowen Du  
Beihang University  
BDBC lab  
dubowen@buaa.edu.cn

Jiefeng Peng  
Sun Yat-sen University  
jiefengpeng@gmail.com

Liang Lin  
Sun Yat-sen University  
linliang@iee.org

## ABSTRACT

Traffic flow prediction is crucial for urban traffic management and public safety. Its key challenges lie in how to adaptively integrate the various factors that affect the flow changes. In this paper, we propose a unified neural network module to address this problem, called Attentive Crowd Flow Machine (ACFM), which is able to infer the evolution of the crowd flow by learning dynamic representations of temporally-varying data with an attention mechanism. Specifically, the ACFM is composed of two progressive ConvLSTM units connected with a convolutional layer for spatial weight prediction. The first LSTM takes the sequential flow density representation as input and generates a hidden state at each time-step for attention map inference, while the second LSTM aims at learning the effective spatial-temporal feature expression from attentionally weighted crowd flow features. Based on the ACFM, we further build a deep architecture with the application to citywide crowd flow prediction, which naturally incorporates the sequential and periodic data as well as other external influences. Extensive experiments on two standard benchmarks (i.e., crowd flow in Beijing and New York City) show that the proposed method achieves significant improvements over the state-of-the-art methods.

## CCS CONCEPTS

• **Computing methodologies** → *Machine learning*; • **Applied computing**;

## KEYWORDS

Mobility Data, Traffic Flow Prediction, Spatial-Temporal Modeling, Memory and Attention Neural Networks

\*Corresponding author is Guanbin Li. This work was supported by the State Key Development Program under Grant 2018YFC0830103, the National Natural Science Foundation of China under Grant 61702565 and Grant 51778033, the Science and Technology Planning Project of Guangdong Province under Grant 2017B010116001, and was also sponsored by CCF-Tencent Open Research Fund.

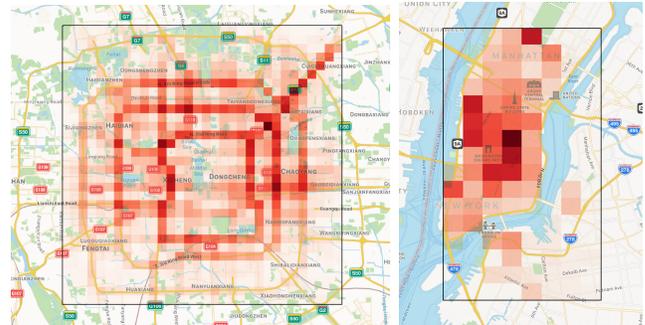
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

MM '18, October 22–26, 2018, Seoul, Republic of Korea

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5665-7/18/10...\$15.00

<https://doi.org/10.1145/3240508.3240636>



**Figure 1: Visualization of two crowd flow maps in Beijing and New York City. We partition a city into a grid map based on the longitude and latitude and generate the crowd flow maps by measuring the number of crowd in each region with mobility data (e.g., GPS signals or mobile phone signals). The weight of each grid indicates the flow density of a time period at a specific area.**

## ACM Reference Format:

Lingbo Liu, Ruimao Zhang, Jiefeng Peng, Guanbin Li, Bowen Du, and Liang Lin. 2018. Attentive Crowd Flow Machines. In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3240508.3240636>

## 1 INTRODUCTION

Crowd flow prediction is crucial for traffic management and public safety, and has drawn a lot of research interests due to its huge potentials in many intelligent applications, including intelligent traffic diversion and travel optimization.

Nowadays, we live in an era where ubiquitous digital devices are able to broadcast rich information about human mobility in real-time and at a high rate, which exponentially increases the availability of large-scale mobility data (e.g., GPS signals or mobile phone signals). In this paper, we generate the crowd flow maps from these mobility data and utilize the historical crowd flow maps to forecast the future crowd flow of a city. As shown in Fig. 1, we partition a city into a grid map based on the longitude and latitude, and measure the number of pedestrians in each region at each time interval with the mobility data. Although the regional scale can vary greatly in different cities, the core problem lies in excavating the evolution of traffic flow in different spatial and temporal regions.

Recently, notable successes have been achieved for citywide crowd flow prediction based on deep neural networks coupled with

certain spatial-temporal priors [35, 36]. Nevertheless, there still exist several challenges limiting the performance of crowd flow analysis in complex scenarios. First, crowd flow data can vary greatly in temporal sequence and capturing such dynamic variations is non-trivial. Second, some periodic laws (e.g., traffic flow suddenly changing due to rush hours or pre-holiday effects) can greatly affect the situation of crowd flow, which increases the difficulty in learning crowd flow representations from data.

To solve all above issues, we propose a novel spatial-temporal neural network module, called Attentive Crowd Flow Machine (ACFM), to adaptively exploit diverse factors that affect crowd flow evolution and at the same time produce the crowd flow estimation map end-to-end in a unified module. The attention mechanism embedded in ACFM is designed to automatically discover the regions with primary positive impacts for the future flow prediction and simultaneously adjust the impacts of the different regions with different weights at each time-step. Specifically, the ACFM comprises two progressive ConvLSTM [31] units. The first one takes input from i) the feature map representing flow density at each moment and ii) the memorized representations of previous moments, to compute the attentional weights, while the second LSTM aims at generating superior spatial-temporal feature representation from attentionally weighted sequential flow density features.

The proposed ACFM has the following appealing properties. First, it can effectively incorporate spatial-temporal information in feature representation and can flexibly compose solutions for crowd flow prediction with different types of input data. Second, by integrating the deep attention mechanism [20, 24], ACFM adaptively learns to represent the weights of each spatial location at each time-step, which allows the model to dynamically perceive the impact of the given area at a given moment for the future traffic flow. Third, ACFM is a general and differentiable module which can be effectively combined with various network architectures for end-to-end training.

In addition, for forecasting the citywide crowd flow, we further build a deep architecture based on the ACFM, which consists of three components: i) sequential representation learning, ii) periodic representation learning and iii) a temporally-varying fusion module. The first two components are implemented by two parallel ACFMs for contextual dependencies modeling at different temporal scales, while the temporally-varying fusion module is proposed to adaptively merge the two separate temporal representation for crowd flow predictions.

The main contributions of this work are three-fold:

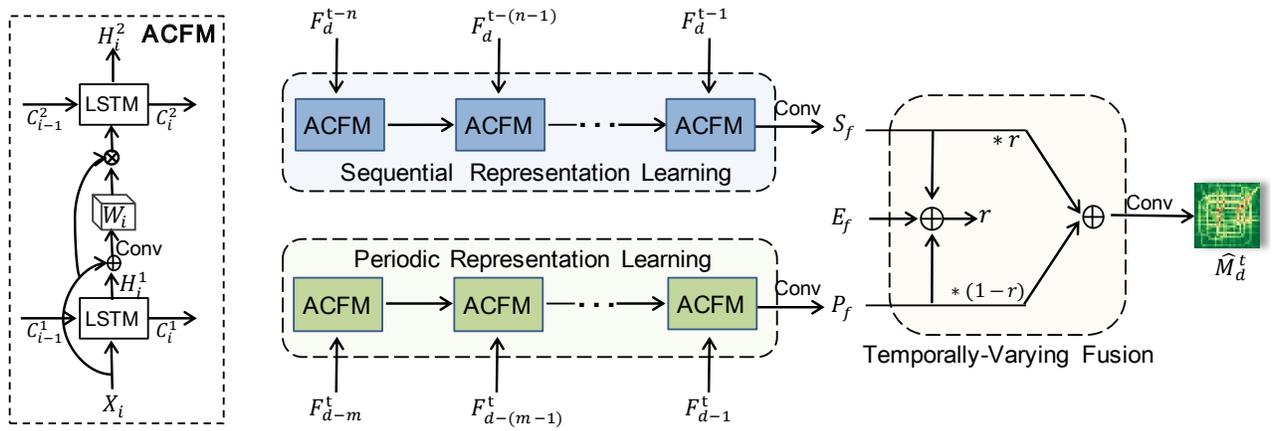
- We propose a novel ACFM neural network, which incorporates two LSTM modules with spatial-temporal attentional weights, to enhance the crowd flow prediction via adaptively weighted spatial-temporal feature modeling.
- We integrate ACFM in our customized deep architecture for citywide crowd flow estimation, which recurrently incorporates various sequential and periodic dependencies with temporally-varying data.
- Extensive experiments on two public benchmarks of crowd flow prediction demonstrate that our approach outperforms existing state-of-the-art methods by large margins.

## 2 RELATED WORK

**Crowd Flow Analysis.** Due to the wide application of traffic congestion analysis and public safety monitoring, citywide crowd flow analysis has recently attracted a wide range of research interest [41]. A pioneer work was proposed by Zheng et al., [40], in which they proposed to represent public traffic trajectories as graphs or tensor structures. Inspired by the significant progress of deep learning on various tasks [5, 17, 18, 37, 38], many researchers also have attempted to handle this task with deep neural network. Fouladgar et al. [9] introduced a scalable decentralized deep neural networks for urban short-term traffic congestion prediction. In [36], a deep learning based framework was proposed to leverage the temporal information of various scales (i.e. temporal closeness, period and seasonal) for crowd flow prediction. Following this work, Zhang et al., [35] further employed a convolution based residual network to collectively predict inflow and outflow of crowds in every region of a city grid-map. To take more efficient temporal modeling into consideration, Dai et al. [6] proposed a deep hierarchical neural network for traffic flow prediction, which consists of an extraction layer to extract time-variant trend in traffic flow and a prediction layer for final crowd flow forecasting. Currently, to overcome the scarcity of crowd flow data, Wang et al. [27] proposed to learn the target city model from the source city model with a region based cross-city deep transfer learning algorithm.

**Memory and attention neural networks.** Recurrent neural networks (RNN) have been widely applied to various sequential prediction tasks [8, 25]. As a variation of RNN, Long Short-Term Memory Networks (LSTM) enables RNNs to store information over extended time intervals and exploit longer-term temporal dependencies. It was first applied to the research field of natural language processing [21] and speech recognition [11], while recently many researchers have attempted to combine CNN with LSTM to model the spatial-temporal information for various of computer vision applications, such as video salient object detection [16], image caption [23, 30] and action recognition [26]. Visual attention is a fundamental aspect of human visual system, which refers to the process by which humans focus the computational resources of their brain's visual system to specific regions of the visual field while perceiving the surrounding world. It has been recently embedded in deep convolution networks [4] or recurrent neural networks to adaptively attend on mission-related regions while processing feed-forward operation and have been proved effective for many tasks, including machine translation [21], crowd counting [19], multi-label image classification [28], face hallucination [3], and visual question answering [33]. However, no existing work incorporates attention mechanism in crowd flow prediction.

The most relevant works to us are [32, 39], which also incorporate ConvLSTM for spatial-temporal modeling. However, they are used for consecutive video frames representation and aims to estimate the crowd counting on a given surveillance image instead of forecasting crowd flow evolution based on mobility data. Moreover, our proposed ACFM is composed of two progressive LSTM modules with learnable attention weights, which is not only adept at modeling spatial-temporal representation, but also efficiently capturing the effect on the global crowd flow evolution caused by the changes of traffic conditions in each particular spatial-temporal



**Figure 2: Left: The architecture of the proposed Attentive Crowd Flow Machine (ACFM).** ACFM can be applied to adequately capture various contextual dependencies for crowd flow evolution analysis.  $X_i$  denotes the input feature map of the  $i^{th}$  iteration. “ $\oplus$ ” denotes feature concatenation and “ $\otimes$ ” refers to element-wise multiplication. **Right: The architecture of our citywide crowd flow prediction networks.** It consists of sequential representation learning, periodic representation learning and a temporally-varying fusion module.  $F_d^t$  denotes the embedding feature of crowd flow and external factors at the  $t^{th}$  time interval of the  $d^{th}$  day.  $\hat{M}_d^t$  is the predicted crowd flow map.  $S_f$  and  $P_f$  are sequential representation and periodic representation, while external factors integrative feature  $E_f$  is the element-wise addition of external factors features of all relative time intervals. The symbols  $r$  and  $(1 - r)$  reflect the importance of  $S_f$  and  $P_f$  respectively.

region (e.g. a traffic jam caused by an accident). Last but not the least, the attention mechanism embedded in our ACFM module also helps to improve the interpretability of the network process while boosting the performance.

### 3 PRELIMINARIES

In this section, we first describe some basic elements of crowd flow and then define the crowd flow prediction problem.

**Region Partition:** There are many ways to divide a city into multiple regions in terms of different granularities and semantic meanings, such as road network [7] and zip code tabular. In this study, following the previous works [34, 35], we partition a city into  $h \times w$  non-overlapping grid map based on the longitude and latitude. Each rectangular grid represents a different geographical region in the city. Figure 1 illustrates the partitioned regions of Beijing and New York City.

**Crowd Flow:** In practical application, we can extract a mass of crowd trajectories from GPS signals or mobile phone signals. With those crowd trajectories, we can measure the number of pedestrians entering or leaving a given region at each time interval, which are called inflow and outflow in our work. For convenience, we denote the crowd flow map at the  $t^{th}$  time interval of  $d^{th}$  day as a tensor  $M_d^t \in R^{2 \times h \times w}$ , of which the first channel is the inflow and the second channel is the outflow. Some crowd flow maps are visualized in Figure 5.

**External Factors:** As mentioned in the previous work [35], crowd flow can be affected by many complex external factors, such as meteorology information and holiday information. In this paper, we also consider the effect of these external factors. The meteorology information (e.g., weather condition, temperature and wind speed) can be collected from some public meteorological websites,

such as Wunderground<sup>1</sup>. Specifically, the weather condition is categorized into sixteen categories (e.g., sunny and rainy) and it is digitized with One-Hot Encoding [12], while temperature and wind speed are scaled into the range [0, 1] with min-max linear normalization. Multiple categories of holiday<sup>2</sup> (e.g., Chinese Spring Festival and Christmas) can be acquired from calendar and encoded into a binary vector with One-Hot Encoding. Finally, we concatenate all external factors data to a 1D tensor. The external factors tensor at the  $t^{th}$  time interval of  $d^{th}$  day is expressed as a  $E_d^t$  in the following sections.

**Crowd Flow Prediction:** This problem aims to predict the crowd flow map  $M_d^t$ , given historical crowd flow maps and external factors data until the  $(t - 1)^{th}$  time interval of  $d^{th}$  day.

### 4 ATTENTIVE CROWD FLOW MACHINE

We propose a unified neural network module, named Attentive Crowd Flow Machine (ACFM), to learn the crowd flow spatial-temporal representations. ACFM is designed to adequately capture various contextual dependencies of the crowd flow, e.g., the spatial consistency and the temporal dependency of long and short term. As shown on the left of Fig. 2, the ACFM is composed of two progressive ConvLSTM [31] units connected with a convolutional layer for attention weight prediction at each time step. The first LSTM (bottom LSTM in the figure) models the temporal dependency through original crowd flow feature embedding (extracted from CNN), the output hidden state of which is concatenated with current crowd flow feature and fed to a convolution layer for weight map inference. The second LSTM (upper LSTM in the figure) is of the same structure as the first LSTM but takes the re-weighted

<sup>1</sup><https://www.wunderground.com/>

<sup>2</sup>The categories of holiday are variational in different datasets.

crowd flow features as input at each time-step and is trained to recurrently learn the spatial-temporal representations for further crowd flow prediction.

For better understanding, we denote the input feature map of the  $i^{th}$  iteration as  $X_i \in R^{c \times h \times w}$ , with  $h$ ,  $w$  and  $c$  representing the height, width and the number of channels. Following [14], the hidden state  $H_i^1 \in R^{c \times h \times w}$  of first LSTM can be formulated as:

$$H_i^1 = \mathbf{ConvLSTM}(H_{i-1}^1, C_{i-1}^1, X_i), \quad (1)$$

where  $C_{i-1}^1$  is the memorized cell state of the first LSTM at  $(i-1)^{th}$  iteration. The internal hidden state  $H_i^1$  is maintained to model the dynamic temporal behavior of the previous crowd flow sequences.

We concatenate  $H_i^1$  and  $X_i$  to generate a new tensor, and feed it to a single convolutional layer with kernel size  $1 \times 1$  to generate an attention map  $W_i$ , which can be expressed as:

$$W_i = \mathbf{Conv}_{1 \times 1}(H_i^1 \oplus X_i, w), \quad (2)$$

where  $\oplus$  denotes feature concatenation and  $w$  is the parameters of the convolutional layer. And  $W_i$  indicates the weights of each spatial location on the feature map  $X_i$ . We further reweigh  $X_i$  with an element-wise multiplication according to  $W_i$  and take the reweighed map as input to the second LSTM for representation learning, the hidden state  $H_i^2 \in R^{c \times h \times w}$  of which can be formulated as:

$$H_i^2 = \mathbf{ConvLSTM}(H_{i-1}^2, C_{i-1}^2, X_i * W_i), \quad (3)$$

where  $*$  refers to the element-wise multiplication.  $h_i^2$  encodes the attention-aware content of current input as well as memorizes the contextual knowledge of previous moments. The output of the last hidden state thus encodes the information of the whole crowd flow sequence, and is used as the spatial-temporal representation for evolution analysis of future flow map. In the next section, we will show how to incorporate the proposed ACFM in our crowd flow prediction framework.

## 5 CITYWIDE CROWD FLOW PREDICTION

We build a deep neural network architecture incorporated with our proposed ACFM to predict citywide crowd flow. As illustrated on the right of Fig. 2, the crowd flow prediction framework consists of three components: (1) sequential representation learning, (2) periodic representation learning and (3) a temporally-varying fusion module. For the first two parts of the framework, we employ the ACFM to model the contextual dependencies of crowd flow at different temporal scales. After that, a temporally-varying fusion module is proposed to adaptively merge the different feature embeddings from each component with the weight  $r$  learned from the concatenation of respective feature representations and the external information. Finally, the merged feature map is fed to one additional convolution layer for crowd flow map inference.

### 5.1 Sequential Representation Learning

The evolution of citywide crowd flow is usually affected by diverse internal and external factors, e.g., current urban traffic and weather conditions. For instance, a traffic accident occurring on a city main road at 9 am may seriously affect the crowd flow of nearby regions in subsequent time periods. Similarly, a sudden rain may seriously affect the crowd flow in a specific region. To deal

with these issues, we take several continuous crowd flow features and their corresponding external factors features as the sequential temporal features, and feed them into our ACFM to recurrently capture the trend of crowd flow in the short term.

Specifically, we denote the input sequential temporal features as:

$$S_{in} = \{F_d^{t-k} | k = n, n-1, \dots, 1\}, \quad (4)$$

where  $n$  is the length of the sequentially related time intervals and  $F_j^i$  denotes the embedding features of the crowd flow and the external factors at the  $i^{th}$  time interval of the  $j^{th}$  day. The extraction of embedding feature  $F_j^i$  will be described in Section 5.4. We apply the proposed ACFM to learn sequential representation from temporal features  $S_{in}$ . As shown on the right of Fig. 2, the ACFM recurrently takes each element of  $S_{in}$  as input and learns to selectively memorize the context of this specific temporally-varying data. The output hidden state of the last iteration is further fed into a following convolution layer to generate a feature representation of size  $c \times h \times w$ , denoted as  $S_f$ , which forms the spatial-temporal feature embedding of the fine-grained sequential data.

### 5.2 Periodic Representation Learning

Generally, there exist some periodicities which make a significant impact on the changes of traffic flow. For example, the traffic conditions are very similar during morning rush hours of consecutive workdays, repeating every 24 hours. Similar with sequential representation learning described in Section 5.1, we take periodic temporal features

$$P_{in} = \{F_d^t | k = m, m-1, \dots, 1\}, \quad (5)$$

to capture the periodic property of crowd flow, where  $n$  is the length of the periodic days. As shown on the right of Fig. 2, we employ ACFM to learn periodic representation with the periodic temporal features  $P$  as input. The hidden output of the last iteration of ACFM is passed through a convolutional layer to generate a representation  $P_f \in R^{c \times h \times w}$ . The  $P_f$  encodes the context of periodic laws, which is essential for crowd flow prediction.

### 5.3 Temporally-Varying Fusion

The future crowd flow is affected by the two temporally-varying representations  $S_f$  and  $P_f$ . A naive method is to directly merge those two representations, however it is suboptimal. In this subsection, we propose a novel temporally-varying fusion module to adaptively fuse the sequential representation  $S_f$  and the periodic representation  $P_f$  of crowd flow with different weight.

Considering that the external factors may affect the importance proportion of two representations, we take the sequential representation  $S_f$ , periodic representation  $P_f$  and the external factors integrative feature  $E_f$  to calculate the fusion weight, where  $E_f$  is the element-wise addition of external factors features of all relative time intervals and will be described in Section 5.4. As shown on the right of Fig. 2, we first concatenate  $S_f$ ,  $P_f$  and  $E_f$  and feed them as input to two fully-connected layers (the first layer has 512 neurons and the second has only one neuron) for fusion weight inference. After a sigmoid function, the temporally-varying fusion module outputs a single value  $r \in [0, 1]$ , which reflects the importance of

the sequential representation  $S_f$ . And  $1 - r$  is treated as the fusion weight of periodic representation  $P_f$ .

We then merge these two temporal representations with different weight and further reduce the feature to two channels (input and output flow) with a linear transformation, which can be expressed as:

$$M_f = \mathcal{T}(r * S_f + (1 - r) * P_f). \quad (6)$$

where  $\mathcal{T}$  is the linear transformation implemented by a convolution layer with two filters. The predicted crowd flow map  $\widehat{M}_d^t \in R^{2 \times h \times w}$  can be computed as

$$\widehat{M}_d^t = \tanh(M_f). \quad (7)$$

where the hyperbolic tangent  $\tanh$  ensures the output values are within  $[-1, 1]^3$ .

## 5.4 Implementation Details

We first detail the method of extracting crowd flow feature as well as external factors feature and then describe our network optimization.

**Crowd Flow Feature:** For the crowd flow map  $M_j^i$  at the  $i^{th}$  time interval of the  $j^{th}$  day, we extract its feature  $F_j^i(M)$  with a customized ResNet [13] structure, which is stacked by  $N$  residual units without any down-sampling operations. Each residual unit contains two convolutional layers followed by two ReLU layers. We set the channel numbers of all convolutional layers as 16 and the kernel sizes as  $3 \times 3$ .

**External Factors Feature:** For the external factors  $E_j^i$ , we extract its feature with a simple neural network implemented by two fully-connected layers. The first FC layer has 256 neurons and the second one has  $16 \times h \times w$  neurons. The output of the last layer is further reshaped to a 3D tensor  $F_j^i(E) \in R^{16 \times h \times w}$ , which is the final feature of  $E_j^i$ .

Finally, we concatenate  $F_j^i(M)$  and  $F_j^i(E)$  to generate the embedding feature  $F_j^i$ , which can be expressed as

$$F_j^i = F_j^i(M) \oplus F_j^i(E), \quad (8)$$

where  $\oplus$  denotes feature concatenation. For the external factors integrative feature  $E_f$  described in Section 5.3, it is the element-wise addition of  $\{E_d^{t-k} | k = n, n-1, \dots, 1\}$  and  $\{E_d^{t-k} | k = m, m-1, \dots, 1\}$ .

**Network Optimization:** We adopt the TensorFlow [1] toolbox to implement our crowd flow prediction network. The filter weights of all convolutional layers and fully-connected layers are initialized by Xavier [10]. The size of a minibatch is set to 64 and the learning rate is  $10^{-4}$ . We optimize our networks parameters in an end-to-end manner via Adam optimization [15] by minimizing the Euclidean loss for 270 epochs with a GTX 1080Ti GPU.

## 6 EXPERIMENTS

In this section, we first conduct experiments on two public benchmarks (e.g., TaxiBJ [36] and BikeNYC [36]) to evaluate the performance of our model on citywide crowd flow prediction. We further

<sup>3</sup>When training, we use Min-Max linear normalization method to scale the crowd flow maps into the range  $[-1, 1]$ . When evaluating, we re-scale the predicted value back to the normal values and then compare with the ground truth.

Model	TaxiBJ	BikeNYC
SARIMA [29]	26.88	10.56
VAR [22]	22.88	9.92
ARIMA [2]	22.78	10.07
ST-ANN [35]	19.57	-
DeepST [36]	18.18	7.43
ST-ResNet [35]	16.69	6.33
Ours	<b>15.40</b>	<b>5.64</b>

**Table 1: Quantitative comparisons on TaxiBJ and BikeNYC using RMSE (smaller is better). Our proposed method outperforms the existing state-of-the-art methods on both datasets with a margin.**

conduct an ablation study to demonstrate the effectiveness of each component in our model.

## 6.1 Dataset Setting and Evaluation Metric

We forecast the inflow and outflow of citywide crowds on two datasets: the TaxiBJ [36] dataset for taxicab flow prediction and the BikeNYC [36] dataset for bike flow prediction.

**TaxiBJ Dataset:** This dataset contains 22,459 time intervals of crowd flow maps with a size of  $2 \times 32 \times 32$ , which are generated with Beijing taxicab GPS trajectory data. The external factors contain weather conditions, temperature, wind speed and 41 categories of holiday. For the fair comparison, we refer to [35] and take the data in the last four weeks as the testing set and the rest as the training set. In this dataset, we set the sequential length  $n$  and the periodic length  $m$  as 3 and 2, respectively. As with ST-ResNet [35], the ResNet described in Section 5.4 is composed of 12 residual units.

**BikeNYC Dataset:** This dataset is generated with the NYC bike trajectory data, which contains 4,392 available time intervals crowd flow maps with the size of  $2 \times 16 \times 8$ . The data of the last ten days are chosen to be the test set. As for external factors, 20 categories of the holiday are recorded. In this dataset, we set the sequential length  $n$  as 5 and the periodic length  $m$  as 7. For a fair comparison with ST-ResNet [35], we also utilize a ResNet described in Section 5.4 with 4 residual units to extract the crowd flow feature.

We adopt Root Mean Square Error (RMSE) as evaluation metric to evaluate the performances of all the methods, which is defined as:

$$RMSE = \sqrt{\frac{1}{z} \sum_{i=1}^z (\widehat{Y}_i - Y_i)^2}, \quad (9)$$

where  $\widehat{Y}_i$  and  $Y_i$  represent the predicted flow map and its ground truth map, respectively.  $z$  indicates the number of samples used for validation.

## 6.2 Comparison with the State of the Art

We compare our method with six state-of-the-art methods, including Auto-Regressive Integrated Moving Average (ARIMA) [2], Seasonal ARIMA (SARIMA) [29], Vector Auto-Regressive (VAR) [22], ST-ANN [35], DeepST [36] and ST-ResNet [35]. For these compared methods, we use the performances provided by Zhang et al. [35] as their results.

Table 1 summarizes the performance of the proposed method and other six methods. On TaxiBJ dataset, our method decreases

Model	RMSE
PCNN	33.44
PRNN-w/o-Attention	32.97
PRNN	32.52
SCNN	17.48
SRNN-w/o-Attention	16.62
SRNN	16.11
SPN-w/o-Fusion	16.01
SPN	15.40

**Table 2: Quantitative comparisons (RMSE) of different variants of our model on TaxiBJ dataset for component analysis.**

the RMSE from 16.69 to 15.40 when compared with current best model, and achieves a relative improvement of 7.7%. Our method also boosts the prediction accuracy on BikeNYC, i.e., decreases RMSE from 6.33 to 5.64. Note that some compared methods, e.g., ST-ANN, DeepST and ST-ResNet, also employ deep learning techniques. Experimental results demonstrate that our proposed ACFM is able to explicitly model the spatial-temporal feature as well as the attention weighting of each spatial influence, which greatly outperforms the state-of-the-art. Some crowd flow prediction maps of our full model on TaxiBJ dataset are shown on the second row of the Fig. 5. As can be seen, our generated crowd flow map is consistently closest to those of the ground-truth, which is accord with the quantitative RMSE comparison.

### 6.3 Ablation Study

Our full model for citywide crowd flow prediction consists of three components: sequential representation learning, periodic representation learning and temporally-varying fusion module. For convenience, we denote our full model as **Sequential-Periodic Network (SPN)** in the following experiments. To show the effectiveness of each component, we implement seven variants of our full model on the TaxiBJ dataset:

- **PCNN:** directly concatenates the periodic features  $P_{in}$  and feeds them to a convolutional layer with two filters followed by  $\tanh$  to predict future crowd flow;
- **SCNN:** directly concatenates the sequential features  $S_{in}$  and feeds them to a convolutional layer followed by  $\tanh$  to predict future crowd flow;
- **PRNN-w/o-Attention:** takes periodic features  $P_{in}$  as input and learns periodic representation with a LSTM layer to predict future crowd flow;
- **PRNN:** takes periodic features  $P_{in}$  as input and learns periodic representation with the proposed ACFM to predict future crowd flow;
- **SRNN-w/o-Attention:** takes sequential features  $S_{in}$  as input and learns sequential representation with a LSTM layer for crowd flow estimation;
- **SRNN:** takes sequential features  $S_{in}$  as input and learns sequential representation with the proposed ACFM to predict future crowd flow;
- **SPN-w/o-Fusion:** directly merges sequential representation and periodic representation with equal weight (0.5) to predict future crowd flow.

**Effectiveness of Spatial Attention:** As shown in Table 2, adopting spatial attention, SRNN decreases the RMSE by 0.51, compared to SRNN-w/o-Attention. For another pair of variants, PRNN with spatial attention has the similar performance improvement, compared to PRNN-w/o-Attention. Fig. 3 and Fig. 4 show some attentional maps generated by our method as well as the residual maps between the input crowd flow maps and their corresponding ground truth. We can observe that there is a negative correlation between the attentional maps and the residual maps. It indicates that our ACFM is able to capture valuable regions at each time step and make better predictions by inferring the trend of evolution. Roughly, the greater difference a region has, the smaller its weight, and vice versa. We can inhibit the impacts of the regions with great differences by multiplying the small weights on their corresponding location features. With the visualization of attentional maps, we can also get to know which regions have the primary positive impacts for the future flow prediction. According to the experiment, we can see that the proposed model can not only effectively improve the prediction accuracy, but also enhance the interpretability of the model to a certain extent.

**Effectiveness of Sequential Representation Learning:** As shown in Table 2, directly concatenating the sequential features  $S$  for prediction, the baseline variant SCNN gets an RMSE of 17.48. When explicitly modeling the sequential contextual dependencies of crowd flow using the proposed ACFM, the variant SRNN decreases the RMSE to 16.11, with 7.8% relative performance improvement compared to the baseline SCNN, which indicates the effectiveness of the sequential representation learning.

**Effectiveness of Periodic Representation Learning:** We also explore different network architectures to learn the periodic representation. As shown in Table 2, the PCNN, which learns to estimate the flow map by simply concatenating all of the periodic features  $P$ , only achieves RMSE of 33.44. In contrast, when introducing ACFM to learn the periodic representation, the RMSE drops to 32.52. This further demonstrates the effectiveness of the proposed ACFM for spatial-temporal modeling.

**Effectiveness of Temporally-Varying Fusion:** When directly merging the two temporal representations with an equal contribution (0.5), SPN-w/o-fusion achieves a negligible improvement, compared to SRNN. In contrast, after using our proposed fusion strategy, the full model SPN decreases the RMSE from 16.11 to 15.40, with a relative improvement of 4.4% compared with SRNN. The results show that the significance of these two representations are not equal and are influenced by various factors. The proposed fusion strategy is effective to adaptively merge the different temporal representations and further improve the performance of crowd flow prediction.

**Further Discussion:** To analyze how each temporal representation contributes to the performance of crowd flow prediction, we further measure the average fusion weights of two temporal representations at each time interval. As shown in the right of Fig. 6, the fusion weights of sequential representation are greater than that of the periodic representation at most time excepting for wee hours. Based on this observation, we can conclude that the sequential representation is more essential for the crowd flow prediction. Although the weight is low, the periodic representation still helps to improve the performance of crowd flow prediction qualitatively

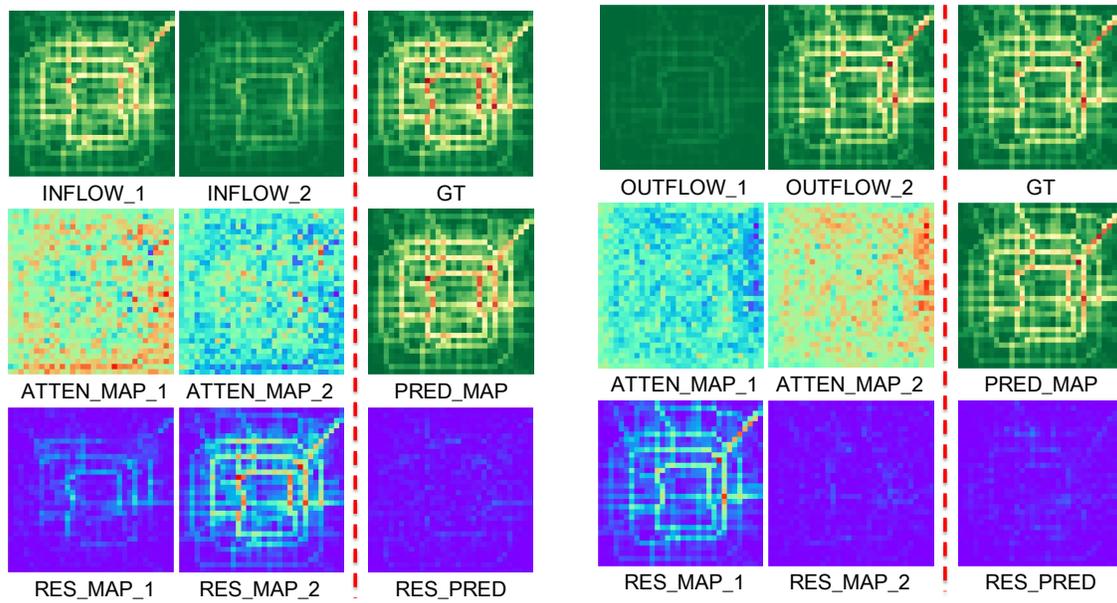


Figure 3: Illustration of the generated attentional maps of the crowd flow in **periodic representation learning** with  $m$  set as 2. Every three columns form one group. In each group: i) on the first row, the first two images are the input periodic inflow/outflow maps and the last one is the ground truth inflow/outflow map of next time interval; ii) on the second row, the first two images are the attentional maps generated by our ACFM, while the last one is our predicted inflow/outflow map; iii) on the third row, the first two images are the residual maps between the input flow maps and the ground truth, while the last one is the residual map between our predicted flow map and the ground truth.

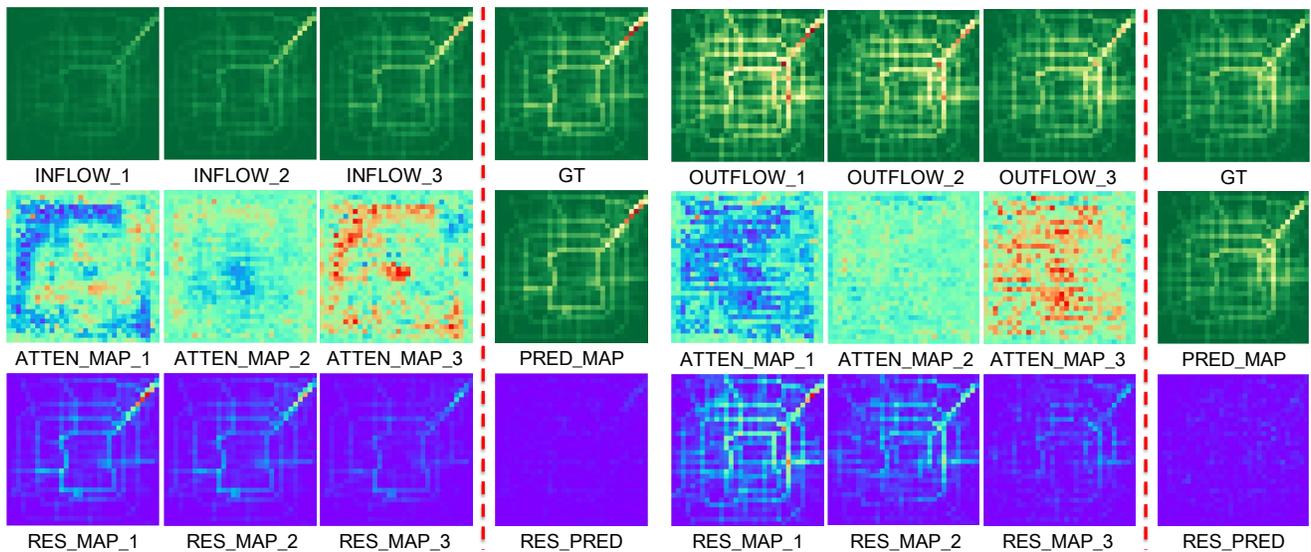
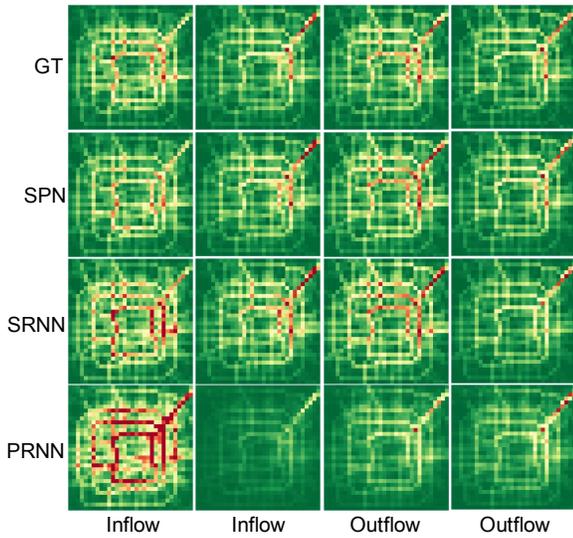


Figure 4: Illustration of the generated attentional maps of the crowd flow in **sequential representation learning** with  $n$  set as 3. Every four columns form one group. In each group: i) on the first row, the first three images are the input sequential inflow/outflow maps and the last one is the ground truth inflow/outflow map of next time interval; ii) on the second row, the first three images are the attentional maps generated by our ACFM, while the last one is our predicted inflow/outflow map; iii) on the third row, the first three images are the residual maps between the input flow maps and the ground truth, while the last one is the residual map between our predicted flow map and the ground truth.



**Figure 5: Visual comparison of predicted flow maps of different variants on TaxiBJ dataset. The first two columns are inflow maps and the other two columns are outflow maps. The first row is the ground truth maps of crowd flow, while the bottom three rows are the predicted flow maps of SPN, SRNN and PRNN respectively. We can observe that i) the combinations of PRNN and SRNN can help to generate more precise crowd flow maps and ii) the difference between the predicted flow maps of our full model SPN and the ground truth maps are relatively small.**



**Figure 6: The average fusion weights of two types of temporal representation on TaxiBJ testing set. We can find that the weights of sequential representation are greater than that of the periodic representation, which indicates the sequential trend is more essential for crowd flow prediction.**

and quantitatively. Fusing with periodic representation, we can decrease the RMSE of SRNN by 4.4% and generate more precise crowd flow maps, as shown in Table 2 and Fig. 5.

## 7 CONCLUSION

This work studies the spatial-temporal modeling for crowd flow prediction problem. To incorporate various factors that affect the flow changes, we propose a unified neural network module named

Attentive Crowd Flow Machine (ACFM). In contrast to the existing flow estimation methods, our ACFM explicitly learns dynamic representations of temporally-varying data with an attention mechanism and can infer the evolution of the future crowd flow from historical crowd flow maps. A unified framework is also proposed to merge two types of temporal information for further prediction. According to the extensive experiments, we have exhaustively verified the effectiveness of our proposed ACFM on the task for citywide crowd flow prediction.

## REFERENCES

- [1] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv:1603.04467* (2016).
- [2] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*.
- [3] Qingxing Cao, Liang Lin, Yukai Shi, Xiaodan Liang, and Guanbin Li. 2017. Attention-aware face hallucination via deep reinforcement learning. *arXiv:1708.03132* (2017).
- [4] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. 2016. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*.
- [5] Tianshui Chen, Liang Lin, Lingbo Liu, Xiaonan Luo, and Xuelong Li. 2016. DISC: Deep Image Saliency Computing via Progressive Representation Learning. *TNNLS* 27, 6 (2016), 1135–1149.
- [6] Xingyuan Dai, Rui Fu, Yilun Lin, Li Li, and Fei-Yue Wang. 2017. DeepTrend: A Deep Hierarchical Neural Network for Traffic Flow Prediction. *arXiv:1707.03213* (2017).
- [7] Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, Linhong Zhu, Rose Yu, and Yan Liu. 2016. Latent Space Model for Road Networks to Predict Time-Varying Traffic. *KDD* (2016), 1525–1534.
- [8] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*.
- [9] Mohammadhani Fouladgar, Mostafa Parchami, Ramez Elmasri, and Amir Ghaderi. 2017. Scalable deep traffic flow neural networks for urban traffic congestion prediction. *arXiv preprint arXiv:1703.01006* (2017).
- [10] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*. 249–256.
- [11] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *ICASSP*.
- [12] David Harris and Sarah Harris. 2010. *Digital design and computer architecture*. Morgan Kaufmann.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.
- [14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [15] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980* (2014).
- [16] Guanbin Li, Yuan Xie, Tianhao Wei, Keze Wang, and Liang Lin. 2018. Flow Guided Recurrent Neural Encoder for Video Saliency Object Detection. In *CVPR*. 3243–3252.
- [17] Ya Li, Lingbo Liu, Liang Lin, and Qing Wang. 2017. Face Recognition by Coarse-to-Fine Landmark Regression with Application to ATM Surveillance. In *CCCV*. Springer, 62–73.
- [18] Lingbo Liu, Guanbin Li, Yuan Xie, Yizhou Yu, and Liang Lin. 2018. Facial Landmark Localization in the Wild by Backbone-Branches Representation Learning. In *BigMM*. IEEE.
- [19] Lingbo Liu, Hongjun Wang, Guanbin Li, Wanli Ouyang, and Liang Lin. 2018. Crowd Counting using Deep Recurrent Spatial-Aware Network. In *IJCAI*.
- [20] Jiasen Lu, Caoming Xiong, Devi Parikh, and Richard Socher. 2016. Knowing when to look: Adaptive attention via A visual sentinel for image captioning. *arXiv:1612.01887* (2016).
- [21] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv:1508.04025* (2015).
- [22] Helmut Lütkepohl. 2011. Vector autoregressive models. In *International Encyclopedia of Statistical Science*. 1645–1647.
- [23] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. 2014. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv:1412.6632* (2014).
- [24] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. 2015. Action recognition using visual attention. *arXiv:1511.04119* (2015).

- [25] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.
- [26] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. 2015. Differential recurrent neural networks for action recognition. In *ICCV*.
- [27] Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. 2018. Crowd Flow Prediction by Deep Spatio-Temporal Transfer Learning. *arXiv preprint arXiv:1802.00386* (2018).
- [28] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. 2017. Multi-Label Image Recognition by Recurrently Discovering Attentional Regions. In *CVPR*.
- [29] Billy Williams, Priya Durvasula, and Donald Brown. 1998. Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models. *Transportation Research Record: Journal of the Transportation Research Board* 1644 (1998), 132–141.
- [30] Xian Wu, Guanbin Li, Qingxing Cao, Qingge Ji, and Liang Lin. 2018. Interpretable Video Captioning via Trajectory Structured Localization. In *CVPR*. 6829–6837.
- [31] SHI Xingjian, Zhouong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NIPS*.
- [32] Feng Xiong, Xingjian Shi, and Dit-Yan Yeung. 2017. Spatiotemporal modeling for crowd counting in videos. In *ICCV*.
- [33] Huijuan Xu and Kate Saenko. 2016. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *ECCV*.
- [34] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, and Jieping Ye. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. *arXiv preprint arXiv:1802.08714* (2018).
- [35] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In *AAAI*. 1655–1661.
- [36] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. 2016. DNN-based prediction model for spatio-temporal data. In *ACM SIGSPATIAL*.
- [37] Ruimao Zhang, Liang Lin, Guangrun Wang, Meng Wang, and Wangmeng Zuo. 2018. Hierarchical Scene Parsing by Weakly Supervised Learning with Image Descriptions. *PAMI* 1 (2018), 1–1.
- [38] Ruimao Zhang, Liang Lin, Rui Zhang, Wangmeng Zuo, and Lei Zhang. 2015. Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *TIP* 24, 12 (2015), 4766–4779.
- [39] Shanghang Zhang, Guanhang Wu, João P Costeira, and José MF Moura. 2017. FCN-rLSTM: Deep Spatio-Temporal Neural Networks for Vehicle Counting in City Cameras. *arXiv:1707.09476* (2017).
- [40] Yu Zheng. 2015. Trajectory data mining: an overview. *TIST* 6, 3 (2015), 29.
- [41] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. *TIST* 5, 3 (2014), 38.