Fine-grained Urban Flow Prediction via a Spatio-Temporal Super-Resolution Scheme

Rujia Shen[1 [0000−0002−0913−3630]], Jian Xu[1✉[0000−0002−0583−5512]], Qing Bao[1 [0000−0002−3550−5098]], Wei Li[1 [0000−0002−0512−9554]], Hao Yuan[1 [0000−0002−4617−591X]], and Ming Xu[1 [0000−0001−9332−5258]]

Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China
{rujiashen, jian.xu, qbao, wei.li, harvey, mxu}@hdu.edu.cn

Abstract. Urban flow prediction plays an essential role in public safety and traffic scheduling for a city. By mining the original granularity flow data, current research methods could predict the coarse-grained region flow. However, the prediction of a more fine-grained region is more important for city management, which means cities could derive more details from the original granularity flow data. In this paper, given the future weather information, we aim to predict the fine-grained region flow. We design Weather-affected Fine-grained Region Flow Predictor (WFRFP) model based on the super-resolution scheme. Our model consists of three modules: 1) *Key flow maps selection* module selects key flow maps from massive historical data as the input instance according to temporal property and weather similarity; 2) *Weather condition fusion* module processes the original weather information and extracts weather features; 3) *Fine-grained flow prediction* module learns the spatial correlations by wide activation residual blocks and predicts the fine-grained region flow by the upsampling operation. Extensive experiments on a real-world dataset demonstrate the effectiveness and efficiency of our method, and show that our method outperforms the state-of-the-art baselines.

Keywords: Deep Learning · Spatio-Temporal Data · Prediction.

1 Introduction

The urban flow prediction system is crucially essential to urban planning, public safety and various applications such as bike-sharing platforms. The current research methods could approximate the future urban flow [22,21]. However, their predicting area is coarse due to the limitation of sparse sensors deployed over the city. Predicting the fine-grained region flow (*e.g.*, crowd flow and traffic flow) under the future weather conditions is important for a city, which can improve traffic management and guarantee public safety.

For instance, as shown in Fig. 1 left, the area is divided into 8×8 grid regions, and we can only obtain the flow of the coarse-grained regions (*e.g.*, R_1) from the sensors deployed over the city. For bike-sharing platforms like Hellobike and Mobike, they launch bikes based on the crowd flow of region R_1. However, region R_1 is large and the crowd flow is unevenly distributed. Magnifying region R_1, we will find that the crowd flow is more concentrated in fine-grained region r_1 (outdoor region) on a sunny day,

but more concentrated in region r_2 (indoor region) on a cloudy day. If people could predict the fine-grained regions flow (*e.g.*, r_2) with future weather conditions, bike-sharing platforms could allocate bike resources reasonably and provide better service for consumers.

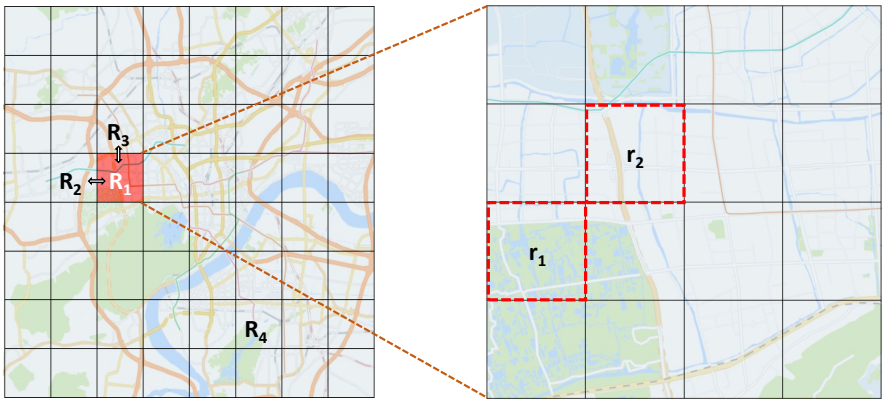

Fig. 1. Left: an area is divided into 8×8 coarse-grained grid region. We can obtain coarse-grained crowd flow of each region by CCTVs and loop detectors. Right: the corresponding fine-grained grid regions of region R_1. We aim to predict the crowd flow of a more precise region (*e.g.*, r_1).

In this paper, our goal is predicting the spatially fine-grained flows from the observed coarse-grained data when the future weather condition is given. Predicting the weather-affected fine-grained region flow means to explore the relationship between the flow distribution and the weather, and reduce the scope of the predicting area to obtain details from coarse-grained flow. However, the prediction is challenging due to the following complex factors:

- **Redundant Input Data.** For the weather-affected fine-grained region flow prediction problem, the input will be a very long historical data sequence, which is difficult for model training. Besides, some historical data is less relevant to the future weather condition, and even bring negative impact for prediction.
- **Spatial Dependencies.** On one side, the flow of a region can be affected by the surrounding regions. A shown in Fig. 1, the outflow from region R_1 affects the region flow of R_2 and R_3. Likewise, outflows from surrounding regions and distant regions (*e.g.*, R_4) will affect the flow of region R_1. On the other side, the corresponding fine-grained flow distribution of a coarse-grained region will change over time and weather conditions.

To tackle these challenges, we propose a deep neural network model, Weather-affected Fine-grained Region Flow Predictor (**WFRFP**), which could predict the fu-

ture fine-grained region crowd flow when the future weather condition is available. WFRFP is inspired by the solutions for single image super-resolution (SISR) problem [3,10,12], which aims at the recovery of a high resolution (HR) image from its low resolution (LR) counterpart. Solutions for SISR problem have motivated applications in other fields, such as satellite [15], medical imaging [17], meteorology [14] and flow inferring [11]. Particularly, Liang et al. [11] firstly proposed the fine-grained urban flow inferring model, which maps coarse-grained flow to fine-grained flow based on super-resolution scheme. Please note that our work is different from the work of Liang et al. [11]. The work of Liang et al. aims to use the flow data that has already been obtained to learn high-resolution mapping, while our work aims to predict the future fine-grained flow from massive historical flow data according to the future weather condition.

In this paper, we adopt the similar idea of SISR and design an appropriate model to predict the weather-affected fine-grained region flow. Specifically, WFRFP first selects weather-similar and time-series data, and then learns weather features and spatial features from these data. Finally, it performs the upsampling operation to obtain the fine-grained region flow. Our contributions are as follows:

- We present the idea to predict the weather-affected fine-grained region flow. And we design the WFRFP model, which can capture weather and spatial impacts on crowd flow, and predict the fine-grained region flow of a more precise region.
- We design a method to select key flow data from massive historical data, which both remains time-series property and has high a correlation with the future weather condition. Our structure reduces the amount of data input while improving the training efficiency.
- We performed experiments and evaluate our framework on a real-world dataset with baselines. And the experimental results proved that our method is superior to other baseline methods.

The rest of this paper is organized as follows. In Section 2, we define notations and formalize the problem of weather-affected fine-grained region flow prediction. In Section 3, we introduce our prediction method in detail. In Section 4, we show the process and the algorithm of model training. In Section 5, we describe our experiment settings and analyze the experimental results. Section 6 reviews the related work and Section 7 concludes our work.

2 Formulation

In this section, we first define notations in this paper and then formulate the problem of Weather-affected Fine-grained Region Flow Prediction (**WFRFP**). We give the notations used in this paper in Table 1.

Definition 1 (Region) As shown in Fig. 1, given the latitude and longitude of an area of interest (*e.g.*, a city, a district, etc.), we divide it into an $I \times J$ grid map. Each grid corresponds to a region [21]. Partitioning the city into smaller regions suggests that we can obtain flow data with more details, which produces a more fine-grained flow map.

Table 1. Notations and meanings

Notations	Meanings
$\mathbf{C}_t$	the coarse-grained flow map at t-th time slot
$\mathbf{F}_t$	the fine-grained flow map at t-th time slot
$\mathbf{M}_t$	the weather condition at t-th time slot
$m_{(q,i)}$	the q-th weather variable of a weather condition at t-th time slot
N	the upsampling factor
$\mathbf{M}^c$	the coarse-grained weather feature map
$\mathbf{M}^f$	the fine-grained weather feature map
$\mathbf{m}_{con}$	vector of continuous weather variables
$\mathbf{m}_{disc}$	vector of discrete weather variables
$\mathcal{C}$	the set of historical coarse-grained flow maps
$\mathcal{M}$	the set of historical weather conditions

Definition 2 (Flow Map) Each divided grid region has a certain region flow, and we define flow maps to describe the flow distribution in grid regions. Let $\mathbf{C}_t \in \mathbb{R}_+^{I\times J}$ represent a coarse-grained crowd flow map at the t-th time slot. $\mathbb{R}_+$ denotes the flow class (*e.g.*, crowd flow, traffic flow, etc.). Given an upsampling factor N, we represent a fine-grained crowd flow map at the t-th time slot as $\mathbf{F}_t \in \mathbb{R}_+^{NI\times NJ}$. Fig. 2 shows an example when upsampling factor N is 2. Each coarse-grained grid in Fig. 2 left is divided into 2×2 fine-grained grids in Fig. 2 right.

Definition 3 (Weather Condition) Let $\mathbf{M}_t$ represents the weather condition at the t-th time slot. Each $\mathbf{M}_t$ contains several weather variables m. According to whether the value of the variable is continuous, we divide those weather variables into two categories: discrete and continuous weather variables. Continuous weather variables include temperature, rainfall and wind speed. Discrete weather variables include weather categories (*e.g.*, sunny, cloudy) and time information. We will process those variables into an embedding vector for model training.

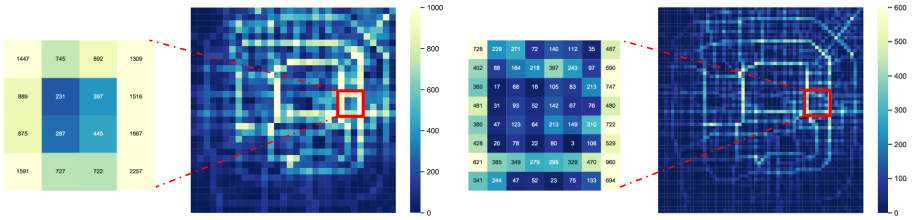

Fig. 2. An example when upsampling factor N is 2. Left: a 32×32 coarse-grained flow map. Right: a 64×64 fine-grained flow map. Each coarse-grained grid is divided into 2×2 fine-grained grids.

Problem Statement (WFRFP) Given the historical coarse-grained region flow maps $\mathcal{C} = \{\mathbf{C}_0, \mathbf{C}_1, \cdots, \mathbf{C}_t\}$, the historical weather conditions $\mathcal{M} = \{\mathbf{M}_0, \mathbf{M}_1, \cdots, \mathbf{M}_t\}$,

the future weather condition $\mathbf{M}_{t+\Delta t}$ and an upsampling factor N, predict the future fine-grained region crowd flow map $\mathbf{F}_{t+\Delta t}$.

3 Methodology of WFRFP Model

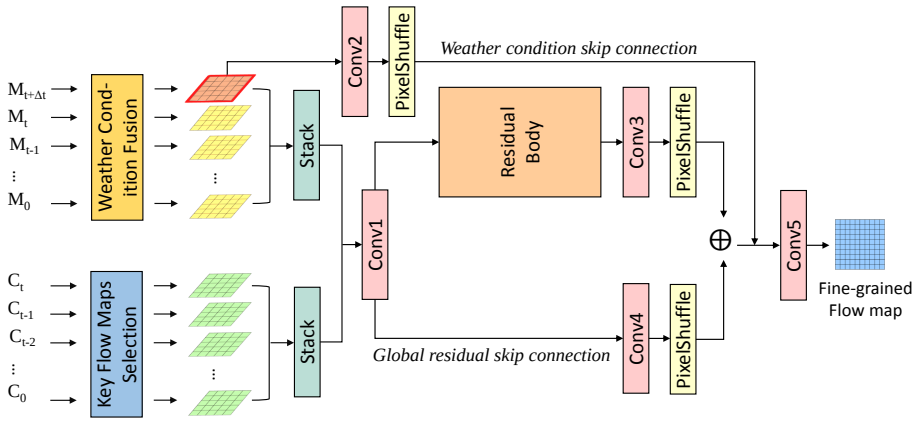

Fig. 3. The architecture of our model. Our model consists of three modules: *key flow maps selection* module, *weather condition fusion* module and *fine-grained flow prediction* module.

Fig. 3 shows the architecture of our model, which consists of the following three parts: key flow maps selection, weather condition fusion and fine-grained flow prediction.

Key flow maps selection. This part aims to select key flow maps from the massive historical data as the input sequence for model training. Our method depicts temporal correlations by time-series method, and takes the impact of the future weather condition into account by selecting weather-similar flow data, which reduces the amount of data input and improves training efficiency.

Weather condition fusion. Original weather condition information cannot be fed to neural networks directly, thus we process original information before model training. We first transform weather variables into low-dimension vectors, and then we obtain weather feature maps though weather feature extraction network. The weather feature maps will be concatenated with the key flow maps as the input for model training.

Fine-grained flow prediction. This module learns the spatial correlations and predicts the fine-grained region flow. We first use wide activation residual blocks to extract spatial correlation features, which allow more low-level information to pass through. Then, we perform the upsampling operation with a global residual skip connection to obtain fine-grained flow maps.

3.1 Key Flow Maps Selection

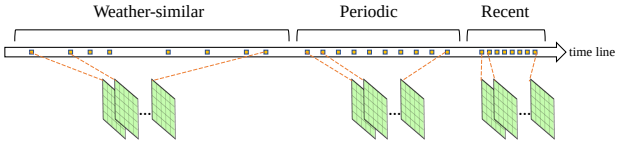

Fig. 4. Method proposed to select key flow maps. We divide the time line into three parts, and select the high-relevant flow maps according to temporal property and weather similarity of each part.

Due to such a long sequence of historical urban flow data, feature learning and model training become very challenging. In addition, there are two kinds of flow maps more relevant and significant than others for prediction: 1) flow maps with similar weather conditions to the future condition; 2) flow maps at similar times to the prediction target. For instance, if tomorrow is rainy and the goal is to predict the region flow at 8 p.m., historical flow maps at 8 p.m. with rainy weather condition own a higher correlation than others for prediction. Therefore, we propose an effective method to select these higher-relevant flow maps to reduce the input size, while the generated input sequence still preserves temporal property. Fig. 4 depicts the method of key flow maps selection.

We first select the key flow maps according to temporal property, and divide them into two categories: recent flow maps $\mathcal{C}^r = \left\{\mathbf{C}_t, \mathbf{C}_{t-1}, \cdots, \mathbf{C}_{t-(l_r-1)}\right\}$, and periodic flow maps $\mathcal{C}^p = \left\{\mathbf{C}_{t+\Delta t-p}, \mathbf{C}_{t+\Delta t-2p}, \cdots, \mathbf{C}_{t+\Delta t-l_p \cdot p}\right\}$. l_r and l_p are the length of recent flow map sequence and periodic flow map sequence separately. p is the period interval (*e.g.*, 24 hours), and $t + \Delta t$ is the predicting target time. The recent flow sequence and periodic flow sequence maintain the temporal property, which captures the mobility trend over the predicting area.

Second, we select the weather-similar flow maps from the remaining historical data. We still select periodic flow maps from the remaining historical data, and each flow map owns corresponding weather condition. We normalize the weather conditions data so that those weather variables are scaled to the same range. Then we calculate the similarity between the future weather condition $\mathbf{M}_{t+\Delta t}$ and the weather conditions of these historical periodic flow maps:

$$sim(\mathbf{M}_{t+\Delta t}, \mathbf{M}_q) = \frac{1}{\sqrt{\sum_{i=1}^{N}\left(m_{(t+\Delta t,i)} - m_{(q,i)}\right)^2}} \\ s.t.\ q = t + \Delta t - k \cdot p, k \in \mathbb{Z}^+. \tag{1}$$

We select the top-k periodic flow maps $\mathcal{C}^s = \{\mathbf{C}_{q_1}, \mathbf{C}_{q_2}, \cdots, \mathbf{C}_{q_k}\}$ with higher sim as the weather-similar flow maps. We will feed weather-similar flow maps $\mathcal{C}^s$ with recent flow maps $\mathcal{C}^r$ and periodic flow maps $\mathcal{C}^p$ as input instance. We obtain the in-

put instance with the consideration of the temporal property and the weather condition similarity, which reduces input size while improving the training efficiency.

3.2 Weather Condition Fusion

Weather conditions have a complicated and significant influence on region flow distribution over the fine-grained regions. For example, even if the total population in a coarse-grained region remains stable over time, under storming weather people tend to move from outdoor regions to indoor regions, which results in the change of flow distribution in the fine-grained regions. Thereby, we design a subnet to handle those implicit weather impacts on flow distribution all at once.

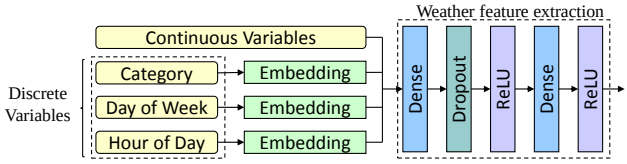

Fig. 5. The structure of weather condition fusion module.

For each flow map selected in Section 3.1, we process corresponding weather conditions inspired by [11], and Fig. 5 shows the architecture of the weather condition fusion module. We have divided weather variables into discrete and continuous variables in Section 2. For continuous weather variables, we directly concatenate them to a vector $\mathbf{m}_{con}$. And for discrete weather variables, we feed them into different embedding layers separately and obtain low-dimensional vectors, and we concatenate those vectors to a vector $\mathbf{m}_{disc}$. Then, the concatenation of $\mathbf{m}_{con}$ and $\mathbf{m}_{disc}$ gives the final embedding vector for weather conditions $\mathbf{m} = [\mathbf{m}_{con}; \mathbf{m}_{disc}]$.

We feed the concatenated vector $\mathbf{m}$ into the weather feature extraction subnetwork whose structure is depicted in Fig. 5. By using dense layers, different external impacts are compounded to construct a hidden representation, which models the complicated interaction. The set of coarse-grained weather feature maps $\mathcal{M}^c$ can be obtained though weather feature extraction subnetwork.

Intuitively, the future weather condition is more influential on the future flow distribution than other historical weather conditions. However, as the neural network goes deeper, future weather information becomes weaker. To avoid the perishing of future weather information, we apply a 3×3 kernel size convolutional layer followed by a PixelShuffle layer [16] to obtain the fine-grained future weather feature map $\mathbf{M}^f$, and $\mathbf{M}^f$ will directly carry the future weather information to the later stage of model training, which plays a similar role as skip connection in Residual Network.

3.3 Fine-grained Flow Prediction

Fine-grained Flow Prediction module extracts high-level features, which describe not only the citywide flow distribution, but also the weather influences, and finally output the fine-grained flow maps.

For SISR problem, it has been demonstrated that with same parameters and computational budgets, models with wider features before activation have significantly better performance [20]. Considering the similarity between SISR and WFRFP, we adopt the wide activation strategy in our network.

After obtaining the key flow maps and weather feature maps as we described in section 3.1 and 3.2, we stack them into a tensor separately. And then the two tensors concatenated together, which is represented by $\mathbf{X}^{(0)}$ will be fed into flow prediction module. Convolutional layer is widely used to capture the correlations between an image pixel and the surrounding pixels. Like pixels in image, the crowd flow of a region is also affected by surrounding regions. So we use a convolutional layer with 5×5 kernel size to extract low-level features. The convolutional layer *Conv1* is defined as follows:

$$\mathbf{X}^{(1)} = f\left(W^{(1)} * \mathbf{X}^{(0)} + b^{(1)}\right) \tag{2}$$

where $*$ denotes the convolution operation, f is an activation function, $W^{(1)}$ and $b^{(1)}$ are the learnable parameters of the convolutional layer *Conv1*. Then we feed the fused feature maps into the residual body, which contains R residual blocks to extract high-level features. Fig. 6 shows the structure of wide activation residual block.

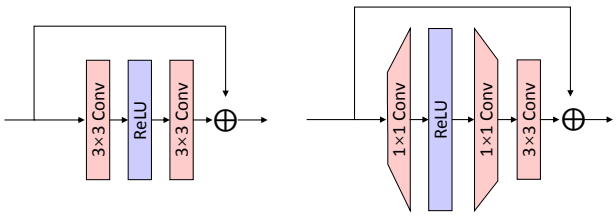

Fig. 6. Left: the structure of vanilla residual block. Right: the structure of residual block with wide activation.

The vanilla residual block (*e.g.*, used in EDSR and MDSR [12]) usually consists of two convolutional layers, *i.e.*, one 3×3 before ReLU activation and another 3×3 after. Both layers produce the same number of feature maps. Different from the vanilla residual block, the wide activation residual block expands the features before activation to let more low-level information pass through ReLU activation and produce more feature maps. Specifically, to reduce the computational cost, we first slim the features of residual identity mapping pathway. Then, a 1×1 convolution layer with more channel numbers (*e.g.*, 256 channels) is used to expand the features. After ReLU, two convolutional layers with fewer features are applied sequentially: one 1×1 convolutional layer

for reducing the number of features to 64 and the other 3×3 convolutional layer for further spatial-wise feature extraction.

Since we utilize a fully convolutional residual architecture, the reception field grows larger as the network goes deeper. In other words, each pixel at the high-level feature map will be able to capture distant or even citywide spatial correlations. After extracting the high-level features though residual blocks, we perform the upsampling operation. We use a 3×3 convolutional layer and a PixelShuffle layer [16] to rear-range and upsample the feature maps to $N\times$ size with the number of channels unchanged, and output $NI \times NJ$ fine-grained flow feature map $\mathbf{F}'$.

We take the low-level flow feature as input, and use the same upsampling structure (a convolution layer followed by a PixelShuffle layer) as the global residual skip connection, which building an information highway skipping over the residual blocks to allow efficient gradient back-propagation. We obtain the fine-grained flow feature map $\mathbf{F}''$ by this way. Finally, we add $\mathbf{F}'$ and $\mathbf{F}''$, and concatenate them with fine-grained weather feature map $\mathbf{M}^f$ obtained in Section 3.2. Then we use a 9×9 convolution layer to map the concatenated feature to a tensor $\mathbf{F}$.

4 Model Training

Our WFRFP model can be trained to predict future fine-grained flow from historical region flow data and weather conditions data by minimizing Mean Squared Error (MSE) between the predicted flow matrix $\hat{\mathbf{F}}_t$ and the true flow matrix $\mathbf{F}_t$:

$$L(\theta) = \left\| \mathbf{F}_t - \hat{\mathbf{F}}_t \right\|_F^2 \tag{3}$$

where θ denotes the set of parameters in WFRFP.

Algorithm 1 outlines the training process. We first construct the training instances from the original flow data and weather conditions data (lines 1-17). Then, our model is trained via backpropagation and Adam [9] (lines 18-23).

5 Experiments

In this section, we evaluate the performance of our WFRFP model. We conduct experiments on a real-world dataset and compare 2 evaluation scores between 4 baselines and our proposed model. Besides, we also discuss the performance of different variants of our model.

5.1 Experimental Setting

Dataset. We evaluate our model on TaxiBJ dataset, which contains weather conditions and crowd flows information. Table 2 shows more details about TaxiBJ dataset.

– Weather Conditions: This part records the weather conditions in Beijing during 7/1/2013 to 10/31/2013, which contains 16 types weather category (*e.g.*, sunny), temperature, wind speed, and holiday information.

Algorithm 1 WFRFP Training Algorithm

Input: Historical flow maps and weather conditions: $\{\mathbf{C}_0, \mathbf{C}_1, \cdots, \mathbf{C}_t\}$, $\{\mathbf{M}_0, \mathbf{M}_1, \cdots, \mathbf{M}_t\}$; lengths of recent, periodic, weather-similar sequences: l_r, l_p, l_s;
future weather condition $\mathbf{M}_{t+\Delta t}$; period: p; target time: $t + \Delta t$; upsampling factor N.
Output: Learned WFRFP model

$\mathcal{Q} \leftarrow \emptyset$
for $i \in [0, l_r - 1]$ **do**
  $\mathcal{Q} \leftarrow \mathbf{C}_{t-i}$;
end for
for $i \in [1, l_p]$ **do**
  $\mathcal{Q} \leftarrow \mathbf{C}_{t+\Delta t-i \cdot p}$;
end for
for $i \in [1, l_s]$ **do**
  Compute $sim(\mathbf{M}_{t+\Delta t}, \mathbf{M}_{t+\Delta t-l_r-p\cdot(l_p+i)})$ with Eq. 1
  if $sim(\mathbf{M}_{t+\Delta t}, \mathbf{M}_{t+\Delta t-l_r-p\cdot(l_p+i)}) < \alpha$ **then**
    $\mathcal{Q} \leftarrow \mathbf{M}_{t+\Delta t-l_r-p\cdot(l_p+i)}$;
  end if
end for
for $i \in [0, length(\mathcal{Q})]$ **do**
  Select corresponding weather conditions and obtain weather feature maps $\mathbf{M}$ though weather feature extraction module;
  Select corresponding $\mathcal{Q} \leftarrow \mathbf{M}$;
end for
initialize all learnable parameters θ in WFRFP
repeat
  randomly select a batch of instances $\mathcal{Q}_b$ from $\mathcal{Q}$
  find θ by minimizing the objective $L(\theta)$ in Eq. 3 with $\mathcal{Q}_b$
until *stopping criteria is met*
return

– Crowd flows: This part records the taxi flows traveling through Beijing during 7/1/2013 to 10/31/2013. The studied area is split into 32×32 grids, where each grid reports the coarse-grained flow information every 30 minutes. The data format is a 32×32 matrix, and each value in the matrix represents the flow of a grid region. Here, we utilize the coarse-grained taxi flows to predict fine-grained flows with 2× resolution (N = 2) and 4× resolution (N = 4).

Training Details and Hyper-parameters. In our experiment, we partition the data into non-overlapping training, validation and test data by a ratio of 2:1:1 respectively. Conv1 uses a convolutional layer with 64 filters of kernel size 5×5. Conv2, Conv3 and Conv4 uses a convolutional layer with 64 filters of kernel size 3×3. Conv5 uses a convolutional layer with 1 filters of kernel size 9×9. The settings of 3 convolutional layers in wide activation residual block are 256 filters of kernel size 1×1, 64 filters of kernel size 1×1 and 64 filters of kernel size 3×3. The number of residual block is set to 16 and the batch size is 16. l_r, l_p and l_s are set to 2, 3 and 3 respectively. The period interval p is 24 hours.

Table 2. Dataset Description.

TaxiBJ	Value
Time span	7/1/2013-10/31/2013
Time interval	30 minutes
Coarse-grained size	32×32
Upsampling factor (N)	2, 4
Fine-grained size	64×64, 128×128
Weather Data	Value
Category	16 types (*e.g.*, Sunny)
Temperature/°C	[-24.6, 41.0]
Wind speed/mph	[0, 48.6]
Holidays	41

Evaluation Metrics. We use RMSE (Root Mean Square Error), MAE (Mean Absolute Error) as the evaluation metrics for each model:

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left\|\mathbf{F}_t - \hat{\mathbf{F}}_t\right\|_2^2} \tag{4}$$

$$MAE = \frac{1}{T}\sum_{t=1}^{T}\left\|\mathbf{F}_t - \hat{\mathbf{F}}_t\right\|_1 \tag{5}$$

where $\mathbf{F}$ and $\hat{\mathbf{F}}_t$ are the ground truth and the prediction of the fine-grained flow map at t-th time slot, and T is the total number of test samples. In general, RMSE favors spiky distributions, while MAE focuses more on the smoothness of the outcome. Smaller metric scores indicate better model performances.

Baselines.

- Linear Interpolation Scaling (LIS): Linear interpolation is widely used in image upsampling. we first utilize linear interpolation to upsample the flow maps and compute the flow ratio of each fine-grained region to corresponding coarse-grained region. Then we obtain the fine-grained flow by the Hadamard product of the flow ratio and coarse-grained flow maps.
- SRCNN [3]: SRCNN presented the first successful introduction of convolutional neural networks (CNNs) into the SISR problems. It consists of three layers: patch extraction, non-linear mapping and reconstruction.
- VDSR [8]: SRCNN has several drawbacks such as slow convergence speed and limited representation ability owing to the three-stage architecture. Inspired by the VGG-net, VDSR model adopts Very Deep neural networks architecture with depth up to 20. This study suggests that a large depth is necessary for the task of SR.
- EDSR [12]: By applying the residual architecture [5] and removing unnecessary modules from residual architecture, EDSR achieve improved results while making model compact. EDSR also employs residual scaling techniques to stably train large models.

– WFRFP-p: To evaluate the key flow maps selection module, we also compare it with WFRFP-p, which trains WFRFP model with only periodic flow map sequences.

5.2 Experiment Result

Table 3. Comparison among baselines and our method for two upsampling factor.

Method	2		4	
	RMSE	MAE	RMSE	MAE
LIS	16.458	6.874	11.652	4.279
SRCNN	12.759	7.686	8.336	4.792
VDSR	11.032	7.105	8.159	4.513
EDSR	10.601	6.930	7.973	4.380
WFRFP	**10.316**	**6.629**	**7.730**	**4.273**
WFRFP-p	11.351	6.775	7.995	4.324

Study on Model Comparison. Table 3 depicts the experiment result of baselines and our method. According to the results, our fine-grained flow prediction model advances baseline methods, which brings $2.7\% \sim 59.6\%$ improvement under two upsampling factors ($N = 2$ and $N = 4$). Moreover, several important observations are as follows:

– All deep neural network approaches have lower RMSE and MAE than the interpolation method, which verifies that convolutional architecture could learn correlations with spatial features by extracting high-level information.
– The framework obtains about $1.2\% \sim 2.2\%$ and $3.4\% \sim 10.0\%$ enhancement on two metrics respectively after employing wide activation residual blocks. Experiment result demonstrates the advantage of expand the features before activation, which could allow more low-level information to pass though and capture spatial correlations more effectively.
– The advance of WFRFP over WFRFP-p proves that key flow maps selection module plays a significant role in our method. Constructing the input instance by selecting high-relevant data could improve the prediction performance and reduce the input size.
– For each model, the performance of $N = 2$ has higher RMSE and MAE than $N = 4$. This may be because the spatial feature becomes complex and vague when the predicting region gets larger. But we lake of different sizes of flow data to confirm the supposition.

Study on Number of Residual Blocks. Fig. 7 depicts the the variation of accuracy and training time with the increase of the number of residual blocks. At first, as the number of residual block increases, the value of RMSE decreases. This is because deep network

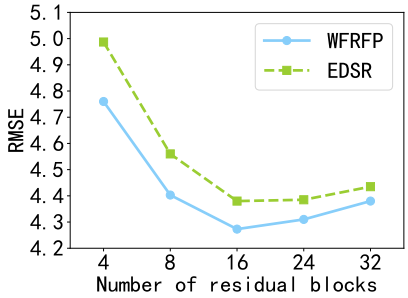

Fig. 7. RMSE Performance on different numbers of residual blocks.

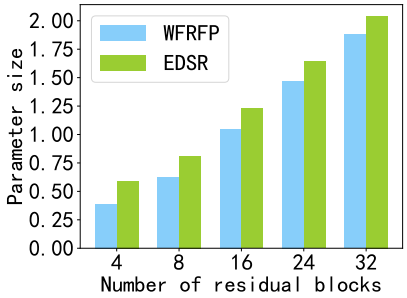

Fig. 8. Parameter size on different numbers of residual blocks.

could capture more distant spatial dependence and learn flow distribution features better. However, when the number of residual block increases to more than 16, the performance of our model starts to decline. The reason is that when the neural network stacks too deep, the probability of model overfitting increases. After many experiments, we found that when the number of residual block is 16, models has the best performance.

Study on Parameter Size. Fig. 8 shows the parameter size of baseline methods and our method as the number of residual blocks grows. The default upsampling factor N is 4. Result shows that WFRFP outperforms baseline method, which demonstrates that our method save more memory space without performance decline by applying wide-activation residual block.

6 Related Work

We review some previous works on urban flow prediction. Urban flow prediction problem is typically spatio-temporal series prediction problems. In earlier time, classical time-series models including Autoregressive Moving Average (ARMA) model [4] and its integrated version such as Vector ARMA model [7] and Integrated Moving Average (ARIMA) model [2] were used to solve urban flow prediction problem. Those classical time-series methods can capture temporal correlations from historical data, however they can not capture the spatial correlations of urban flow distribution.

Later, some machine learning methods were applied to forecast urban flow. Support vector machines have greater generalization ability and guarantee global minima for given training data, and it has proved that SVR performs well for urban flow prediction [18]. Alternatively, STW-KNN [19] is an improved KNN (K-nearest neighbor algorithm) model which enhances forecasting accuracy of urban flow based on the spatio-temporal correlation.

Recently, due to the powerful expressive capabilities of deep neural networks, researchers have started to focus on predicting urban flow by deep learning methods.

Deep-ST [22] firstly utilized a convolutional network to capture spatial correlations and forecasted region flow. Further, ST-ResNet [21] was proposed by employing an advanced residual neural network instead of the general convolution operation. By combining the pyramidal ConvGRU model [1] with periodic representations, Periodic-CRN [23] was designed to model the periodic nature of crowd flow explicitly. STRCNs [6] and DeepSTN+ [13] explored the combination of convolution and LSTM to predict long-term urban flow. All of these deep-learning-based methods have noticed the temporal and spatial dependence of urban flow and tried to find better structures to depict it. However, none of them constructed a special structure for fine-grained flow prediction to capture more details from flow maps.

In this paper, inspired by the solutions for SISR problem, we construct a new structure for fine-grained flow prediction with future weather conditions.

7 Conclusion

In this paper, we formalize the Weather-affected Fine-grained Region Flow Prediction problem and propose WFRFP model, which adopts the idea used in image super-resolution and predict the flow of a more precise region range. Moreover, WFRFP addresses the two challenges, *i.e.*, redundant input data and the spatial dependence by selecting high-relevant flow data and applying the fully convolutional residual architecture. The experiments on the real-world dataset confirm that our method is more effective than other baseline methods for fine-grained region flow prediction. In the future, we will explore the solution for long-term fine-grained flow prediction and take more data sources (*e.g.*, point of interest) into consideration.

Acknowledegments

This work is supported by the National Natural Science Foundation of China (No. 61572165) and the National Natural Science Foundation of China (No. 61806061).

References

1. Ballas, N., Yao, L., Pal, C., Courville, A.: Delving deeper into convolutional networks for learning video representations. arXiv preprint arXiv:1511.06432 (2015)
2. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time series analysis: forecasting and control. John Wiley & Sons (2015)
3. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence **38**(2), 295–307 (2015)
4. Hamilton, J.D.: Time series analysis. Economic Theory. II, Princeton University Press, USA pp. 625–630 (1995)
5. He, K., Zhang, X., Ren, S., Jian, S.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
6. Jin, W., Lin, Y., Wu, Z., Wan, H.: Spatio-temporal recurrent convolutional networks for city-wide short-term crowd flows prediction. In: Proceedings of the 2nd International Conference on Compute and Data Analysis. pp. 28–35. ACM (2018)

7. Kamarianakis, Y., Prastacos, P.: Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches. Transportation Research Record **1857**(1), 74–84 (2003)
8. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
10. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4681–4690 (2017)
11. Liang, Y., Ouyang, K., Jing, L., Ruan, S., Liu, Y., Zhang, J., Rosenblum, D.S., Zheng, Y.: Urbanfm: Inferring fine-grained urban flows. arXiv preprint arXiv:1902.05377 (2019)
12. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 136–144 (2017)
13. Lin, Z., Feng, J., Lu, Z., Li, Y., Jin, D.: Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 1020–1027 (2019)
14. Liu, N., Ma, R., Wang, Y., Zhang, L.: Inferring fine-grained air pollution map via a spatiotemporal super-resolution scheme. In: Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers. pp. 498–504 (2019)
15. Peled, S., Yeshurun, Y.: Superresolution in mri: application to human white matter fiber tract visualization by diffusion tensor imaging. Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine **45**(1), 29–35 (2001)
16. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1874–1883 (2016)
17. Thornton, M.W., Atkinson, P.M., Holland, D.: Sub-pixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using super-resolution pixel-swapping. International Journal of Remote Sensing **27**(3), 473–491 (2006)
18. Wu, C.H., Ho, J.M., Lee, D.T.: Travel-time prediction with support vector regression. IEEE transactions on intelligent transportation systems **5**(4), 276–281 (2004)
19. Xia, D., Wang, B., Li, H., Li, Y., Zhang, Z.: A distributed spatial–temporal weighted model on mapreduce for short-term traffic flow forecasting. Neurocomputing **179**, 246–263 (2016)
20. Yu, J., Fan, Y., Yang, J., Xu, N., Wang, Z., Wang, X., Huang, T.: Wide activation for efficient and accurate image super-resolution. arXiv preprint arXiv:1808.08718 (2018)
21. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
22. Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X.: Dnn-based prediction model for spatio-temporal data. In: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. p. 92. ACM (2016)
23. Zonoozi, A., Kim, J.j., Li, X.L., Cong, G.: Periodic-crn: A convolutional recurrent model for crowd density prediction with recurring periodic patterns. In: IJCAI. pp. 3732–3738 (2018)