# Periodic Residual Learning for Crowd Flow Forecasting

Chengxin Wang
cwang@comp.nus.edu.sg
National University of Singapore

Yuxuan Liang
yuxliang@comp.nus.edu.sg
National University of Singapore

Gary Tan
gtan@comp.nus.edu.sg
National University of Singapore

## ABSTRACT

Crowd flow forecasting, which aims to predict the crowds entering or leaving certain regions, is a fundamental task in smart cities. One of the key properties of crowd flow data is periodicity: a pattern that occurs at regular time intervals, such as a weekly pattern. To capture such periodicity, existing studies either fuse the periodic hidden states into channels for networks to learn or apply extra periodic strategies to the network architecture. In this paper, we devise a novel periodic residual learning network (PRNet) for a better modeling of periodicity in crowd flow data. Unlike existing methods, PRNet frames the crowd flow forecasting as a periodic residual learning problem by modeling the variation between the inputs (the previous time period) and the outputs (the future time period). Compared to directly predicting crowd flows that are highly dynamic, learning more stationary deviation is much easier, which thus facilitates the model training. Besides, the learned variation enables the network to produce the residual between future conditions and its corresponding weekly observations at each time interval, and therefore contributes to substantially more accurate multi-step ahead predictions. Extensive experiments show that PRNet can be easily integrated into existing models to enhance their predictive performance.

## CCS CONCEPTS

• **Applied computing** → **Transportation**; • **Information systems** → **Spatial-temporal systems**.

## KEYWORDS

Crowd flow, periodic residual, spatio-temporal data mining, urban computing, deep learning, convolutional neural networks

**ACM Reference Format:**
Chengxin Wang, Yuxuan Liang, and Gary Tan. 2022. Periodic Residual Learning for Crowd Flow Forecasting. In *The 30th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '22), November 1–4, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3557915.3560947

## 1 INTRODUCTION

Nowadays, the development of intelligent transportation systems has drawn increasing attention as the number of vehicles grows over the years. The total number of motor vehicles has reached 273 million in the U.S. [26], and 6.08 million in Beijing by 2018, respectively, and it has grown to 6570 thousand in Beijing by 2020 [10]. To manage citywide transportation more efficiently, crowd forecasting aims to divide a city into multiple regions, i.e., even grid cells, and generate future vehicles' in/out-flow for each region. It is a crucial task that facilitates a wide range of applications in urban areas, such as assisting transportation managers to alleviate the congestion [39], guiding carsharing companies to pre-allocate vehicles [4], and helping travelers' decision-making [14].

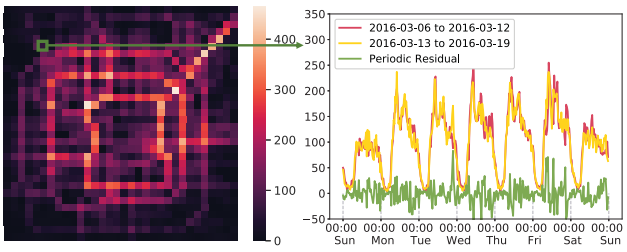


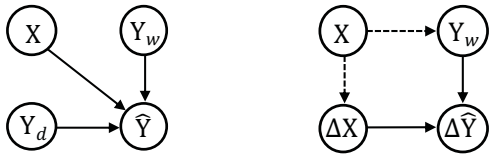

**Figure 1: A visualization of crowd flows in Beijing. Left hand side: the city is divided into many regions; right hand side: the crowd inflow of a region during a period of two weeks, i.e., from 06 March 2016 to 19 March 2016.**

Spatio-temporal (ST) dependency [21, 38] is an important characteristic in crowd flow forecasting: one region's future crowd flow volume is conditioned on other regions' histories and its historical observations. Mainstream works [18, 37] employ convolutional neural networks (CNNs) to capture spatial correlations and utilize different sub-branches or channels to model temporal dependencies of different time scales. Besides, there are some methods that adopt recurrent neural networks (RNNs) [23, 40, 43] or Transformer [28, 34] to enhance temporal modeling via recurrent state transformations or attention mechanisms. However, these models always require higher computational costs and larger storage compared to their CNN counterparts. Meanwhile, more recent works [16, 17] suggest that CNNs can effectively model the spatial and channel-wise correlations simultaneously with the Squeeze-and-Excitation (SE) mechanism [9]. With advanced mechanisms to express complex ST features, prior works have achieved promising prediction results.

Another key characteristic in citywide crowd flow is periodicity [22, 36, 38]. As can be observed from Fig. 1, crowd flow data show periodic patterns, e.g., daily and weekly. For instance, on the daily scale, the volume in the region follows a similar trend that increases during the morning and decreases during the night; on the weekly scale, the flow pattern trends to repeat every week (see the red and yellow line). Existing works on representing such periodic patterns can be summarized in Fig. 2 (a). In detail, the multi-scale time intervals, such as the recent segment, daily segments, and/or weekly segments, are fed into the network for periodic learning. These models can be grouped into two categories - *feature-based*


*SIGSPATIAL '22, November 1–4, 2022, Seattle, WA, USA*

ACM ISBN 978-1-4503-9529-8/22/11.
https://doi.org/10.1145/3557915.3560947



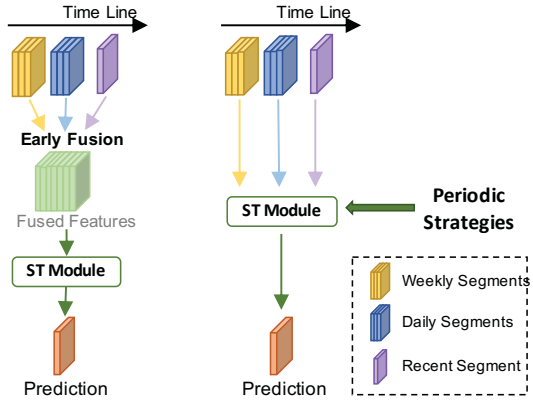

(a) Existing periodic models (b) Ours

**Figure 2: Graphical models for periodic modeling, where X and $\hat{Y}$ represent the current segment and the target segment, respectively. $Y_d$ and $Y_w$ denote segments for the daily scale and the weekly scale, respectively. The solid line indicates the direct relationship, and the dashed line denotes the indirect relationship.**

and *architecture-based* models. As shown in Fig. 3 (a), the feature-based models view the multi-scale observations as different features and concatenate them as a tensor [17, 18] for the network to process. However, the periodic information is mixed in the early stage, while being eliminated as the network depth increases. To tackle this issue, the architecture-based models represent the periodicity more naturally via some extra periodic strategies (see Fig. 3 (b)). For example, DeepST [38] introduces different branches to capture the periodicity; Periodic-CRN [43] designs a loop-back mechanism to integrate the recurrent periodic representations; STDN [35] utilizes the attention mechanism to calculate the similarity of ST representations between multi-scale segments. However, these architectures inevitably induce high computational overheads and extra parameters, which may be prohibitive in large-scale crowd flow forecasting tasks. Considering these facts, one may ask: *can we address the periodic pattern in a more efficient manner?*

To answer this question, we first investigate the inherent periodic behavior of crowd data. As shown in Fig. 1, though the daily crowd flow often fluctuates, the volume difference of a certain region at the same time in successive weeks (we term it as *periodic residual*) tends to be stable even in long-term trends (see the green line). As opposed to raw crowd flow data, periodic residual holds clearer patterns that are easier to learn [2]. We argue that periodic residual features are also consolidated representations extracted from raw data that can help to reduce the difficulties in modeling complex crowd flow patterns. By learning such features, the network can be trained more efficiently, even with fewer parameters. Based on this insight, we propose to think from a new perspective - introducing the residual concept to represent the periodic behavior.

In this paper, we present a novel architecture-based framework entitled **P**eriodic **R**esidual **Net**work (PRNet) for multi-step ahead crowd flow forecasting. Instead of designing complex ST extraction models or sophisticated periodic strategies, PRNet focuses on learning periodic residuals. As depicted in Fig. 2 (b), PRNet converts the learning focus from directly generating predictions to computing the periodic residual. Formally, it structures a residual mapping that predicts the future temporal difference based on the past variation. Then, the periodic learning structure in PRNet allows the network to: 1) alleviate the computational costs by representing the periodicity with an efficient differencing function; 2) reduce memory consumption by learning the variation based on one periodic scale (e.g., weekly scale); 3) reduce redundant trainable




(a) Feature-based model (b) Architecture-based model

**Figure 3: Periodicity representation in ST neural networks.**

parameters by encoding each periodic time interval into a shared parameter encoder; 4) make the network more effective and robust in long-term forecasting as the model generates predictions based on the learned periodical residual features at each time interval. To evaluate our periodic learning structure, we further integrate it into different baseline networks (e.g., DeepST [38], ST-ResNet [37], and DeepLGR [17]) and conduct extensive experiments on the real-world datasets. Furthermore, we notice that existing works are inefficient to capture the global ST correlations, and therefore introduce a lightweight ST enhanced network, named Spatial-Channel Enhanced (SCE) Encoder to jointly encode the most salient global spatial correlations as well as channel dependencies, i.e., ST representation. Our main contributions are summarized as follows:

- We devise a simple yet effective periodic residual learning structure that learns the periodic residual at each time interval to improve the accuracy of multi-step ahead prediction. This structure can be easily integrated into existing models.
- We introduce a lightweight Spatial-Channel Enhanced (SCE) Encoder to better capture global spatio-temporal dependencies.
- Experiments on two real-world datasets demonstrate that PRNet surpasses the state-of-the-art approaches in long-term predictions. We also show that the periodic residual learning structure brings significant improvements in performance for existing models, especially under the small data budget.

## 2 RELATED WORK

**Grid-based Crowd Flow Forecasting**. Crowd flow forecasting has been investigated for more than four decades. Early attempts employ statistical models [1, 2, 7] to predict the condition of crowd flows. In particular, some works [25, 31] investigate the periodicity in crowd flows and apply the seasonal ARIMA to model it. However, these works rely on assumptions of linearity and stationarity and thereby cannot model the complex nonlinear ST dependency. Recently, deep learning models [4, 17, 37] have been used to capture the complex ST correlations. For example, DeepST [38] and ST-ResNet [37] adopt CNN-based architectures to learn ST correlations and achieve higher forecasting accuracy. Specifically, they

**Table 1: The notations of crowd flow, where $P$ refers to the total number of selected periods and $p$ is the periodic index.**

| Notation | Symbol | Definition | Color in Fig 4 | Shape |
|---|---|---|---|---|
| Closeness | $\mathbf{X}_c$ | Current segment | Purple | $H \times W \times 2 \times T_{obs}$ |
| Periodic closeness | $\mathbf{X}_p$ | Periodic observations to the current segment | Blue | $H \times W \times 2 \times T_{obs}$ |
| Prediction | $\mathbf{Y}$ | Target segment for prediction | Green | $H \times W \times 2 \times T_{pred}$ |
| Periodic prediction | $\mathbf{Y}_p$ | Periodic observations to the target segment | Orange | $H \times W \times 2 \times T_{pred}$ |
| Closeness residual | $\Delta\mathbf{X}$ | The residual between closeness and each periodic closeness | Pink | $P \times H \times W \times 2 \times T_{obs}$ |
| Prediction residual | $\Delta\mathbf{Y}$ | The residual between prediction and each periodic prediction | Brown | $P \times H \times W \times 2 \times T_{pred}$ |

integrate the periodicity into networks by feeding multi-scale segments to different sub-branches. For better periodic representations, other works model the periodic pattern explicitly by looping back the periodic representation dictionary [43] or learning the temporal similarity [35]. However, they need massive computation costs to loop back recurrent hidden states or compute attention scores. Recent efforts focus on improving spatial modeling for more accurate forecasts. Graph neural networks (GNNs) [12, 29] have become the frontier of spatial interactions learning in road-based networks [5, 11, 41], however, they have not demonstrated the advantages over CNNs on grid-based problems. Unlike the road-based network that is naturally a non-Euclidean graph, the even grid cells in the grid-based task are treated as pixels, without an explicit graph structure. Meanwhile, CNNs have adequate ability to fully learn spatial interactions between grids via spatial kernels of each layer. Recently, Liang et al. [17] shows CNNs can effectively capture the ST correlations by jointly modeling spatial correlations and temporal dynamics. Besides grid-based crowd flow forecasting, there are some works on predicting on irregular regions [24].

**CNNs and Attention Mechanisms**. CNNs have been successfully applied to many domains, such as computer vision [6], audio generation [27], crowd flow prediction [37], etc. Recent works [8, 9] utilize gating and attention mechanisms to further enhance the feature interdependencies in CNNs. Specifically, SENet [9] introduces a squeeze-and-excitation (SE) operation as the gating mechanism to recalibrate the channel-wise attention through the sigmoid function. However, the global average pooling in SE suppresses spatial information, which makes the network fail to capture spatial correlations effectively. Although some works further introduce attention to enhance the spatial representation via operating additional convolutions layers on average- and max-pooled features [32] or employing dilated convolutions to enlarge the receptive field [20], they fail to fully uncover the global correlations. DANet [3] captures global ST dependencies by extending the self-attention to position attention and channel attention. However, it is computationally expensive since it takes all spatial information into account. In this paper, we model the global ST representation in a computationally efficient manner by only considering the most salient features.

## 3 FORMULATION

In this section, we first define some notations and then formulate the problem of crowd flow forecasting.

**Definition 1 (Region)**: As shown in Fig. 1, we follow [13, 15, 19, 37] to evenly partition an area of interest (such as a city) into $H \times W$ regions, i.e., grid cells, based on their longitude and latitude.

**Definition 2 (Crowd flow)**: The crowd flows at a certain time $\tau$ can be denoted as a 3D tensor $\mathcal{P}^\tau \in \mathbb{R}^{H \times W \times D}$, where $D$ is the number of attributes, e.g., inflow/outflow. Given a region $(h, w)$, *inflow* refers to the total number of incoming traffic entering this region from other regions during a given time interval, while *outflow* is the total number of outcoming traffic leaving this region.

**Definition 3 (Closeness & Periodic closeness)**: For better illustration, we define several segments in Table 1 and Fig. 4. Given the current timestamp $\tau$, the recent segment (i.e., closeness [37]) and its corresponding periodic segments (i.e., periodic closeness in Table 1) are denoted as:

$$\mathbf{X}_c = \mathcal{P}^{\tau - T_{obs}:\tau} = \left[\mathcal{P}^{\tau - T_{obs}}, \cdots, \mathcal{P}^{\tau}\right],$$

$$\mathbf{X}_{1:P} = \left[\mathcal{P}_1^t, \mathcal{P}_2^t \cdots, \mathcal{P}_P^t\right]_{t=\tau - T_{obs} - l*p}^{\tau - l*p},$$

where $T_{obs}$ is the length of recent observations, $P$ refers to the total number of selected periods, $l$ denotes the length of period, and $p$ is the period index. See more details in Table 1 and Fig. 4.

**Definition 4 (Prediction & Periodic prediction)**: After introducing closeness, we represent the target segment for prediction at time $\tau$ and its corresponding periodic segments as:

$$\mathbf{Y} = \mathcal{P}^{\tau+1:\tau+Tpred} = \left[\mathcal{P}^{\tau+1}, \cdots, \mathcal{P}^{\tau+T_{pred}}\right],$$

$$\mathbf{Y}_{1:P} = \left[\mathcal{P}_1^t, \mathcal{P}_2^t \cdots, \mathcal{P}_P^t\right]_{t=\tau+1-l*p}^{\tau+T_{pred}-l*p},$$

where $T_{pred}$ is the length of target predictions.

**Definition 5 (Closeness residual & Prediction residual)**: *Closeness residual* denotes the residuals between $\mathbf{X}_c$ and $\mathbf{X}_{1:P}$, and *prediction residual* represents the residuals between $\mathbf{Y}$ and $\mathbf{Y}_{1:P}$ as:

$$\Delta\mathbf{X} = \left[\mathbf{X}_c - \mathcal{P}_1^t \cdots, \mathbf{X}_c - \mathcal{P}_P^t\right]_{t=\tau - T_{obs} - l*p}^{\tau - l*p},$$

$$\Delta\mathbf{Y} = \left[\mathbf{Y} - \mathcal{P}_1^t \cdots, \mathbf{Y} - \mathcal{P}_P^t\right]_{t=\tau+1-l*p}^{\tau+T_{pred}-l*p},$$

**Problem Statement (crowd flow forecasting):** Given closeness $\mathbf{X}_c$, periodic closeness $\mathbf{X}_{1:P}$, periodic prediction $\mathbf{Y}_{1:P}$, the goal is to

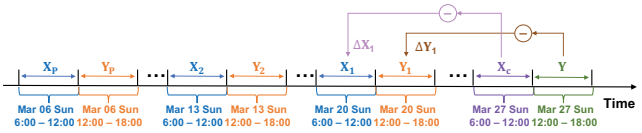



**Figure 4: An example of the multi-scale segments notation under weekly scale, where the length of period $l$ is one week.**

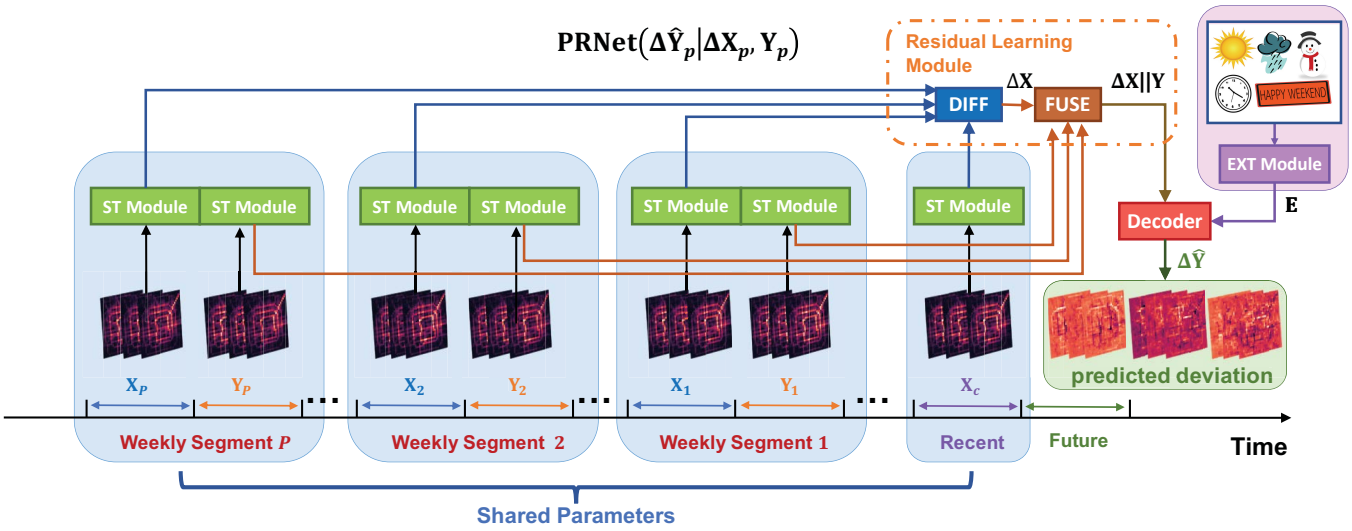



**Figure 5: The overview of PRNet, where ST Module captures the ST correlations of each observed segment simultaneously. Then the network employs a differencing function (DIFF) to provide the closeness residual, and a fusion function (FUSE) to generate representations for the prediction residual. The decoder generates predicted deviations for all periodical weeks.**

predict the prediction residual $\Delta\hat{\mathbf{Y}}$, which is equivalent to predict the future crowd flows $\hat{\mathbf{Y}}$.

## 4 PERIODIC RESIDUAL LEARNING

Fig. 5 illustrates the pipeline of our proposed PRNet, whose core is a periodic residual learning structure. With the structure, PRNet reduces the data non-stationary by utilizing the closeness residual to assist prediction residual generation. For each segment (i.e., closeness, periodic closeness, and periodic prediction), we first fed the raw inputs to the shared ST Module for spatio-temporal representation. Once we obtain the high-level features for each segment, we utilize a Residual Learning Module to learn prediction residual features. These features are then used to generate the predicted deviations via a Decoder. The details of PRNet will be elaborated in the following sections.

### 4.1 Spatio-Temporal (ST) Module

Generality is one of the advantages of our proposed model. Most of the existing Spatio-Temporal (ST) networks can be easily integrated into PRNet as the Spatio-Temporal (ST) Module.

A variety of ST networks has been designed to capture spatio-temporal dependencies. Based on the learning strategy, we group them into two categories, i.e., joint ST learning network and factorized ST learning network. As its name suggests, joint ST learning networks simultaneously capture spatial and temporal dependencies by mapping the temporal inputs to CNN channels and utilizing the CNN kernels for spatio-temporal dependencies extraction [17, 37]. In contrast, factorized ST learning networks decompose the modeling of ST into two separate dimensions, i.e., spatial dimension and temporal dimension. More specifically, they capture the spatial interactions and temporal dependencies sequentially via convolutional layers [35] or convolutional graph layers [39] for spatial dimension and recurrent mechanisms [23] or attention mechanisms [39] for temporal dimension. Among these two schemes, joint ST learning networks are usually applied to grid-based crowd flow forecasting for two reasons: 1) The grid cells in the tasks are even and can be treated as pixels. 2) Recurrent and attention operations usually require high computation costs, especially when the multi-scale time intervals need to be considered [18].

In PRNet, ST Module extracts high-level spatio-temporal representations (denoted as $\mathbf{h}$) for each segment via the ST network:

$$\mathbf{h} = f(\mathcal{P}^{t:t+T_{obs}}; \mathbf{W}_{st}), \tag{1}$$

where $\mathbf{h} \in \mathbb{R}^{H\times W\times C}$ is the output features; $f$ represents the function of an ST network; $\mathbf{W}_{st}$ denotes the learnable parameters, and $t$ is the start timestep of a given time interval. Unlike existing attempts that encode multi-scale time intervals into compacted features [17, 18], each segment in our PRNet is fed separately to a shared ST Module to save parameter usage. More details of our proposed ST Module will be introduced in Section 5.

### 4.2 Residual Learning Module

Statistical methods [2, 30] have demonstrated robust prediction by removing trends and seasonality given time series data. In light of these approaches, we introduce a similar concept to deep learning networks by devising a residual learning module to eliminate the sequential seasonality (i.e., periodicity in this paper). This new module aims to learn the high-level features of the periodic residual that are less complex but still maintain the periodic information. It consists of two functions: differencing function and fusion function.

**Differencing function (DIFF)** removes the seasonality and provides the periodic closeness residual as a reference of temporal shifting to the network. Traditional statistical approaches use the subtraction function to eliminate the seasonality [2]. Thus, we also

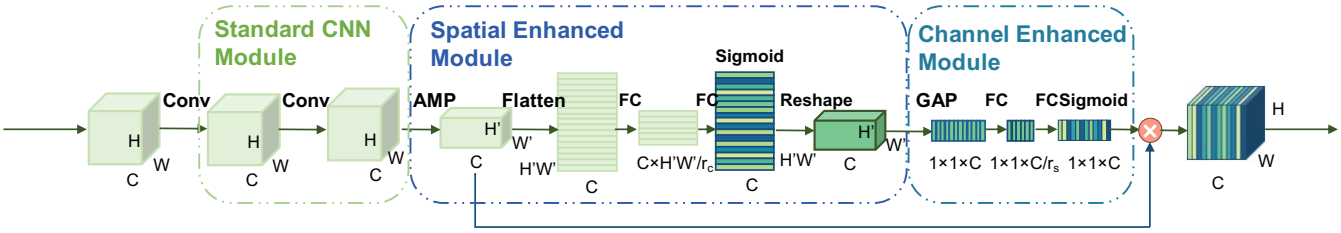



**Figure 6: An illustration of Spatial-Channel Enhanced (SCE) Block in SCE Encoder.**

choose it as our differencing operation since the learned ST features from the ST Module map to their corresponding raw observations. Then the hidden states of periodic closeness residual can be calculated by subtracting the hidden state of closeness $\mathbf{h}_x$ from the hidden state of periodic closeness $\mathbf{h}_{px}$ generated by ST Module:

$$\nabla_d \mathcal{H} = \mathbf{h}_x - \mathbf{h}_{px}, \tag{2}$$

where $\nabla_d$ denotes the differencing operator, and $\nabla_d \mathcal{H} \in \mathbb{R}^{P \times H \times W \times C}$. Note that dimension broadcast is used.

**Fusion function (FUSE)** generates the prediction residual features for the Decoder to produce the prediction residual (i.e., residuals between future crowd flows $\mathbf{Y}$ and its corresponding periodic predictions $\mathbf{Y}_p$). As the periodic closeness residual and the periodic predictions can provide the time-shifting references for the prediction residual, we consider such information into the network by adopting a concatenation function followed by a canonical linear layer on their features:

$$\tilde{\mathcal{H}} = \mathbf{W}_d(\nabla_d \mathcal{H} \parallel \mathbf{h}_{py}) \tag{3}$$

where $\parallel$ is the concatenation operation, and $\mathbf{W}_d$ denotes learnable parameters. Therefore, the embedded vector $\tilde{\mathcal{H}} \in \mathbb{R}^{P \times H \times W \times C}$ can represent the hidden states of the prediction residual, which are conditioned on the features of closeness residual and periodic prediction. It enables the model to learn the deviations between future conditions and its historical observations. It is worth noting that with the residual learning strategy, PRNet provides more stationary features to the network so that it increases the model capacity with no extra costs in parameter space.

## 4.3 External Module & Decoder

External factors, such as date, event, and weather, can affect crowd flows [16, 37, 42]. The External (EXT) Module works on encoding these factors and outputs the external factor embedding $\mathbf{E}$. Same as the ST Module, the EXT Module in PRNet is also a general module, which can be plugged by any existing attempts [37, 38] or be omitted [17]. The same form is for the Decoder. And the default Decoder of PRNet is a fully-connected layer (FC). However, instead of generating absolute values, PRNet focuses on fully uncovering the temporal shifting in periodicity by predicting the variation $\Delta\hat{\mathbf{Y}}$ between the future and its corresponding historical average flows based on $\tilde{\mathcal{H}}$ or the concatenation of $\tilde{\mathcal{H}}$ and $\mathbf{E}$. To strengthen the robustness of our model, all $P$ historical segments are considered. Therefore, we define the loss function as:

$$\mathcal{L}(\theta) = \sum_{\tau=1}^{T_{pred}} \left\| \Delta\hat{\mathbf{Y}}^{\tau} - \Delta\mathbf{Y}^{\tau} \right\|_1, \tag{4}$$

where $\theta$ denotes learnable parameters in the model. Then the predicted deviation $\Delta\hat{\mathbf{Y}} \in \mathbb{R}^{P \times H \times W \times 2 \times T_{pred}}$ can be easily converted to the absolute crowd flows $\hat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times 2 \times T_{pred}}$:

$$\hat{\mathbf{Y}} = \sum_{i=1}^{P} (\Delta\hat{\mathbf{Y}} + \mathbf{Y}_p)/P, \tag{5}$$

where $P$ is the total number of the periodic segments.

# 5 ST MODULE - SCE ENCODER

In Section 4.1, we have discussed that joint ST learning networks are widely used as the backbone to capture the spatio-temporal features of grid-based crowd flows. Among them, many works [18, 37] focus on modeling long-range spatial correlations. Because the one grid cell's crowd flow can be affected by distant neighbors due to vehicles moving very far given a large time interval. To learn long-range spatial dependencies, existing works stack multiple CNN layers to larger the receptive field [37, 38] or apply fully-connected layers to aggregate the features from all grids [18]. However, they have underestimated the temporal relationship between channels within feature maps. Recently, Liang et al. [17] adopts squeeze-and-excitation networks (SENet) [9] to explicitly model the channel-wise relations to enhance ST representation learning. However, it fails to capture complex global patterns as it squeezes global spatial features at each block. Although it further introduces a pyramid CNN structure to abstract the spatial features at different levels for global spatial learning, it also requires additional computational costs. To address the above issue, we propose a Spatial Channel Enhanced (SCE) Encoder as the ST Module in our proposed PRNet. It enhanced the SENet by introducing a lightweight global spatial enhanced module to emphasize the global salient spatial features. The SCE Encoder contains two main components: Embedding Layer and Spatial-Channel Enhanced Block.

## 5.1 Embedding Layer

We follow the previous studies [18, 37] to employ an embedding layer for a feature transformation. In detail, this layer converts each observed segment $\mathcal{P} \in \mathbb{R}^{H \times W \times D \times T}$ to feature maps $\mathbf{z} \in \mathbb{R}^{H \times W \times C}$ through a convolutional operation with kernel size 1, where $T$ denotes the total time intervals of the segment, namely $T_{obs}$ for closeness and $T_{pred}$ for prediction.

## 5.2 Spatial-Channel Enhanced Block

In Fig. 6, we illustrate a single SCE block in SCE Encoder. It comprises three main modules: Standard CNN Module, Spatial Enhanced Module, and Channel Enhanced Module. Since the spatial and temporal information has been indexed to dimensions and

channels, the Standard CNN Module can capture the local spatio-temporal correlation via convolution layers:

$$\vec{\mathbf{h}}^{(m)} = \mathbf{W}_{f2}^{(m)} \star \left(\delta\left(\mathbf{W}_{f1}^{(m)} \star \mathbf{h}^{(m)}\right) + b_{f1}^{(m)}\right) + b_{f2}^{(m)} \tag{6}$$

where $\mathbf{W}_{f1}$, $\mathbf{W}_{f2}$, $b_{f1}$ and $b_{f2}$ are learnable parameters, $\star$ refers to a convolution operator, $\delta(\cdot)$ is ReLU activation function, and $m$ denotes index number of SCE Blocks. Note that $\mathbf{h}^{(0)}$ is $\mathbf{z}$ and $\vec{\mathbf{h}} \in \mathbb{R}^{H \times W \times C}$. We will omit the index $m$ for the same block in the following sections.

**Spatial Enhanced Module (SEM)** enhances the standard CNN by selecting the salient features globally for better spatial representation. To achieve it, we adopt adaptive max pooling (AMP) to down-sample the hidden state $\vec{\mathbf{h}}$ by selecting most important features $\tilde{\mathcal{S}} \in \mathbb{R}^{H' \times W' \times C}$ and translate it to $\mathcal{S}' \in \mathbb{R}^{C \times H'W'}$. Then the excitation operator [9] is adopted to adaptively recalibrate these global salient features for better spatial correlation modelling:

$$\hat{\mathbf{h}}_s = \sigma(g(\mathcal{S}', \mathbf{W}_s)) = \sigma\left(\delta\left(\mathcal{S}'\mathbf{W}_{s1}\right)\mathbf{W}_{s2}\right), \tag{7}$$

where $\sigma$ refers to sigmoid function, $\delta$ denotes the ReLU function, $g(\cdot)$ represents the gated function, $\hat{\mathbf{h}}_s \in \mathbb{R}^{C \times H'W'}$, $\mathbf{W}_{s1} \in \mathbb{R}^{H'W' \times r_s}$, $\mathbf{W}_{s2} \in \mathbb{R}^{r_s \times H'W'}$, and $r_s \ll H'W'$. By using learnable parameters $\mathbf{W}_{s1}$ and $\mathbf{W}_{s2}$ to reduce and increase the feature dimensions sequentially, the gated function enables the network to dynamically control the bypass signals and only capture the most salient features. Then we reshape $\hat{\mathbf{h}}_s$ and obtain the final encoded global spatial feature $\widetilde{\mathbf{h}}_s \in \mathbb{R}^{C \times H' \times W'}$.

**Channel Enhanced Module (CEM)** Except for spatial correlations, dynamic spatio-temporal dependencies need to be considered in crowd flow tasks. We thus propose to use CEM to learn spatial correlations and temporal dependencies simultaneously for better ST understanding. It first summarizes the global spatial features into a channel descriptor, then the descriptor captures the spatio-temporal correlations based on the channel dimension. We adopt global average pooling (GAP) to squeeze the global spatial features and generate channel-wise statistics:

$$\mathbf{c} = \frac{1}{H' \times W'} \sum_{h=1}^{H'} \sum_{w=1}^{W'} \widetilde{\mathbf{h}}_s(h, w), \tag{8}$$

where $\mathbf{c} \in \mathbb{R}^C$. Then a similar strategy as Eq. 7 is used to enhance the spatio-temporal representation by producing the compacted channel-wise features:

$$\widetilde{\mathbf{h}} = \sigma(g(\mathbf{c}, \mathbf{W}_c)) = \sigma\left(\mathbf{W}_{c2}\delta\left(\mathbf{W}_{c1}\mathbf{c}\right)\right) \tag{9}$$

where $\mathbf{W}_{c1} \in \mathbb{R}^{\frac{C}{r_c} \times C}$, $\mathbf{W}_{c2} \in \mathbb{R}^{C \times \frac{C}{r_c}}$, $r_c$ is the reduction ratio, and $\widetilde{\mathbf{h}} \in \mathbb{R}^{1 \times 1 \times C}$. The final output of one SCE Block can be obtained by scaling the compacted features $\widetilde{\mathbf{h}}^{(m)} \in \mathbb{R}^{1 \times 1 \times C}$ and the ST feature map $\vec{\mathbf{h}}^{(m)}$:

$$\mathbf{h}^{(m+1)} = \widetilde{\mathbf{h}}^{(m)} \vec{\mathbf{h}}^{(m)}. \tag{10}$$

By stacking multiple SCE blocks, SCE Encoder can model long-term spatio-temporal dependencies effectively. We stack a total number of $M$ SCE blocks in the SCE Encoder. The receptive field of succeeding blocks in SCE Encoder is larger than the receptive field of former blocks. Therefore, our model constructs simple direct ST interactions between grids in former blocks and indirect global ST connections in the succeeding blocks. To this end, SCE Encoder can efficiently describe correlations between grids over time.

# 6 EXPERIMENTS

## 6.1 Experimental Settings

*6.1.1 Datasets.* We conduct experiments on two real-world datasets [38], i.e., TaxiBJ and BikeNYC. TaxiBJ dataset is the crowd flow dataset, which is obtained through taxicab GPS data. It comprises four sub-datasets - P1, P2, P3, and P4. And BikeNYC dataset records the bike trajectory information which is extracted from the NYC bike system. The detailed statistical information of the datasets is described in Table 2. Besides, the external features of the datasets include holidays, weather conditions, temperature, and wind speed.

**Table 2: The statistic of TaxiBJ and BikeNYC dataset.**

| Dataset | Grid Map | Time Interval (mm/dd/yyyy) | Time Span | Min - Max Value |
|---|---|---|---|---|
| TaxiBJ-P1 | (32, 32) | 07/01/2013 - 10/31/2013 | 30 mins | 0 - 1230 |
| TaxiBJ-P2 | (32, 32) | 03/01/2014 - 06/30/2014 | 30 mins | 0 - 1292 |
| TaxiBJ-P3 | (32, 32) | 03/01/2015 - 06/30/2015 | 30 mins | 0 - 1274 |
| TaxiBJ-P4 | (32, 32) | 11/01/2015 - 04/10/2016 | 30 mins | 0 - 1250 |
| BikeNYC | (16, 8) | 04/01/2014 - 30/09/2014 | 60 mins | 0 - 267 |

**Table 3: The details of data samples over two datasets.**

| Dataset | Days | Total Timeslots | Missing Ratio | Sample Size | | |
|---|---|---|---|---|---|---|
| | | | | Train | Valid | Test |
| TaxiBJ-P1 | 121 | 5808 | 15.8% | 2164 | 720 | 720 |
| TaxiBJ-P2 | 119 | 5712 | 16.3% | 2080 | 693 | 693 |
| TaxiBJ-P3 | 122 | 5596 | 4.4% | 2665 | 887 | 887 |
| TaxiBJ-P4 | 162 | 7776 | 7.2% | 3507 | 1168 | 1168 |
| BikeNYC | 183 | 8784 | 50.0% | 1101 | 366 | 366 |

More details of the two datasets are listed in Table 3. We observe that most datasets have a high missing ratio. To address the missing ratio issue, there are two widely used strategies: Strategy 1 removes the segments with missing values [37, 38], but significantly reducing the number of samples; Strategy 2 fills the missing values with zero to produce a larger data budget, while it introduces extremely noisy data. As the crowd flow shows periodic patterns, we take advantage of the periodicity to address this problem. Specifically, we use a dictionary to store a default periodic value for each time slot and use it to fill the missing value. In the experiments, we set the default value as the mean of known values at the same time slot every week for an affordable way. For example, the default value for Mon 7:00 is the average of the known values on Mon 7:00. The missing value occurring in weekly segments (i.e., periodic closeness, and periodic prediction) will be filled with the default value. And samples with the missing value in the target segment for prediction are discarded. We employ the last 20% data as the test set, and randomly select the remaining 60% data as the training set and 20% as the validation set, respectively. For a fair comparison, the same data preprocessing strategy is adopted for the models, including baselines models.

*6.1.2 Evaluation Metrics.* Following the previous studies [18, 37], we evaluate our model using two metrics: **Mean Absolute Error** (MAE) and **Root Mean Squared Errors** (RMSE).

**Table 4: Model comparison on the TaxiBJ dataset in terms of performance and parameter size, where K denotes thousand and M denotes million. The format of numerical results is "mean ± standard deviation" (the lower results are better).**

| Method | # Params | P1 | | P2 | |
|---|---|---|---|---|---|
| | | MAE | RMSE | MAE | RMSE |
| HA | - | 16.91 | 31.49 | 13.65 | 23.97 |
| DeepST | 380K | 15.68 ± 0.43 | 26.69 ± 0.79 | 15.61 ± 0.35 | 25.48 ± 0.56 |
| ST-ResNet | 3077K | 13.84 ± 0.13 | 23.48 ± 0.16 | 13.74 ± 0.42 | 22.87 ± 0.57 |
| ConvLSTM | 1839K | 11.77 ± 0.06 | 20.19 ± 0.14 | 12.47 ± 0.14 | 21.89 ± 0.36 |
| DeepSTN+ | 105M | 13.41 ± 0.28 | 25.51 ± 0.46 | 12.69 ± 0.45 | 24.03 ± 1.88 |
| Graph WaveNet | 1296K | 12.37 ± 0.05 | 21.07 ± 0.16 | 13.18 ± 0.22 | 23.00 ± 0.40 |
| DeepLGR | 968K | 13.82 ± 0.18 | 25.84 ± 0.45 | 12.09 ± 0.06 | 21.48 ± 0.08 |
| PRNet (Ours) | 711K | **11.76 ± 0.02** | **20.19 ± 0.04** | **12.01 ± 0.02** | **21.12 ± 0.05** |
| **Method** | **# Params** | **P3** | | **P4** | |
| | | MAE | RMSE | MAE | RMSE |
| HA | - | 14.98 | 29.22 | 19.33 | 40.66 |
| DeepST | 380K | 14.94 ± 0.17 | 25.11 ± 0.14 | 15.31 ± 0.35 | 27.45 ± 0.74 |
| ST-ResNet | 3077K | 13.35 ± 0.10 | 23.36 ± 0.32 | 13.39 ± 0.16 | 24.54 ± 0.02 |
| ConvLSTM | 1839K | 12.40 ± 0.03 | 22.12 ± 0.05 | 12.07 ± 0.10 | 23.70 ± 0.23 |
| DeepSTN+ | 105M | 12.21 ± 0.02 | 21.89 ± 0.23 | 12.22 ± 0.11 | 24.15 ± 0.34 |
| Graph WaveNet | 1296K | 13.40 ± 0.16 | 23.98 ± 0.19 | 13.24 ± 0.21 | 25.58 ± 0.54 |
| DeepLGR | 968K | 12.19 ± 0.06 | 21.91 ± 0.15 | 12.39 ± 0.14 | 24.09 ± 0.25 |
| PRNet (Ours) | 711K | **12.09 ± 0.02** | **21.70 ± 0.04** | **11.90 ± 0.05** | **23.25 ± 0.13** |

*6.1.3 Implementation Details.* Our model is trained on a single GTX 2080 Ti using Adam optimizer with a learning rate of 0.0005. We set $T_{obs}$ to 12, $T_{pred}$ to 12, $D$ to 2, $C$ to 64, and $M$ to 9. The convolution kernel size in $\mathbf{W}_{f1}, \mathbf{W}_{f2}$ is $3 \times 3$ with 64 filters. $H'$ and $W'$ are set to 8. $r_s$ and $r_c$ are 8 and 4, respectively. We apply a scalar with 50 on taxi volume. The early-stop strategy is applied in all the experiments. The maximum epoch is set to 250.

*6.1.4 Baselines.* We compare our model with seven baselines:

- **HA** is a traditional time series method that averages the historical flow of the same time slot of the same day given past weekly segments.
- **DeepST** [38] is the first deep learning-based approach for grid-based crowd flow prediction, which utilizes convolution operators to extract local spatial correlations and different CNN branches to capture temporal dependencies.
- **ST-ResNet** [37] further enhances DeepST by introducing residual structure to improve the prediction accuracy.
- **ConvLSTM** [23] integrates the convolution operation to RNN structure to enhance the long-term ST relationship modeling.
- **DeepSTN** [18] uses ordinary convolutions and fully-connected layers to capture the local and long-range spatial features, respectively.
- **Graph WaveNet** [33] utilizes graph neural network to learn self-adaptive spatial interaction and employ stacked dilated casual convolutions to capture long sequence dependency.
- **DeepLGR** [17] adopts SE mechanisms to capture spatial correlation and temporal dynamics concurrently.

## 6.2 Experimental Results and Analysis

Table 4 and Table 5 show the prediction results of baselines and our model on two datasets. The results show that our model consistently outperforms existing methods on all datasets. From the results, we can observe that: 1) Traditional methods can outperform deep learning approaches on some datasets, indicating that periodic information is an important characteristic for crowd flow prediction. For example, HA surpasses DeepST and ST-ResNet in P2. The reason is that P2 has a 16.3% missing ratio which causes the size of training samples to be relatively small so that deep models become overfitted. However, HA achieves inferior performance under a larger data budget (e.g. P1, P3, and P4) as it is a nonparametric model, which ignores the ST correlation and the time trend. Our model takes advantage of traditional methods by integrating explicit periodic knowledge to guide the network, and therefore achieves the best performance among all methods across all datasets. 2) ConvLSTM, DeepSTN, Graph WaveNet, and DeepLGR show better results compared to DeepST and ST-ResNet, which demonstrates better ST correlation understanding can lead to better performance. We notice that DeepLGR can outperform Graph WaveNet, even though Graph WaveNet has a stronger temporal network, i.e., causal convolution network. We think this is because spatial correlations are fully learned in CNNs via spatial kernels of each layer, while they are predefined in GNNs. 3) Our method achieves superior performance over Graph WaveNet, DeeSTN, DeepLGR. Specifically, it reduces MAE error by 8.49%, 5.48 %, 5.41 % on average on the

**Table 5: Model comparison on BikeNYC dataset.**

| Method | # Params | MAE | RMSE |
|---|---|---|---|
| HA | - | 3.38 | 7.52 |
| DeepST | 143K | 3.75 ± 0.06 | 7.50 ± 0.10 |
| ST-ResNet | 2841K | 3.60 ± 0.02 | 7.32 ± 0.03 |
| ConvLSTM | 1839K | 3.69 ± 0.07 | 8.20 ± 0.21 |
| DeepSTN | 1594K | 3.58 ± 0.05 | 7.72 ± 0.07 |
| Graph WaveNet | 1296K | 3.97 ± 0.04 | 8.20 ± 0.11 |
| DeepLGR | 878K | 3.30 ± 0.03 | 7.57 ± 0.09 |
| PRNet (Ours) | 711K | **3.27 ± 0.01** | **7.08 ± 0.02** |

TaxiBJ dataset with 1.82, 147.68, and 1.36 times fewer parameters. On the BikeNYC dataset, it also achieves competitive results with fewer parameters. Note that the parameter numbers of DeepST, ST-ResNet, DeepSTN, and DeepLGR on TaxiBJ and BikeNYC are different. Because their region-specific design for external feature encoding or spatial feature extraction leads to parameter growth as grid cells grow, especially for DeepSTN where fully-connect layers are used to capture global spatial dependencies. Surprisingly, ConvLSTM can produce favorable results. The reason could be that the gate mechanism of it helps the model to capture better temporal dependencies in multi-step ahead prediction. However, it also requires high memory usage. Instead, our model utilizes a periodic residual learning strategy to provide stationary features to increase network capacity with fewer parameters. Overall, our work beats all the methods, proving that the network with a well-designed periodic learning strategy can make good crowd flow predictions.

**Table 6: Model w/ PRNet vs. w/o PRNet on TaxiBJ dataset.**

| **Method** | **# Params** | **P1** | **P2** |
|---|---|---|---|
| DeepST | 380K | 15.68 ± 0.43 | 15.61 ± 0.35 |
| DeepST+ | 369K | 13.17 ± 0.14 | 12.69 ± 0.09 |
| ST-ResNet | 3077K | 13.84 ± 0.13 | 13.74 ± 0.42 |
| ST-ResNet+ | 2503K | 11.95 ± 0.07 | 12.38 ± 0.03 |
| DeepLGR | 968K | 13.82 ± 0.18 | 12.09 ± 0.06 |
| DeepLGR+ | 893K | 11.78 ± 0.02 | 12.05 ± 0.06 |
| **Method** | **# Params** | **P3** | **P4** |
| DeepST | 380K | 14.94 ± 0.17 | 15.31 ± 0.35 |
| DeepST+ | 369K | 12.71 ± 0.03 | 13.88 ± 0.04 |
| ST-ResNet | 3077K | 13.35 ± 0.10 | 13.39 ± 0.16 |
| ST-ResNet+ | 2503K | 12.32 ± 0.04 | 12.21 ± 0.02 |
| DeepLGR | 968K | 12.19 ± 0.06 | 12.39 ± 0.14 |
| DeepLGR+ | 893K | 12.10 ± 0.06 | 11.97 ± 0.02 |

## 6.3 Study on Periodic Residual Learning

We further verify the effectiveness of our periodic residual learning structure on different baselines in Table 6. We use **DeepST+**, **ST-ResNet+** and **DeepLGR+** to denote the model using our proposed periodic residual learning structure with the ST network adopted from DeepST, ST-ResNet and DeepLGR, respectively. In other words, we adopt the backbone of DeepST, ST-ResNet, and DeepLGR as ST Module in PRNet. Also, we keep the EXT Module and the Decoder the same as their original networks. As compared in Table 6, the networks with our proposed structure outperform their original models in terms of accuracy and robustness. The proposed structure assists them to reduce the MAE error by 14.77%, 10.52 %, 5.13%, and to promote the robustness (i.e., reduce the standard deviation) by 76.92%, 80.25%, 63.64% on average on TaxiBJ dataset. The parameters of models are also reduced. Because the proposed structure encodes each observed segment with a shared ST Module rather than feeding them into different branches or concatenating them as a tensor. By applying the shared network to each time interval, PRNet provides explicit periodic references to the target segment from its corresponding periodic segments. Thus, it aids the model to increase the accuracy and robustness in long-term prediction, even with fewer parameters. In summary, these results demonstrate the generality of our periodic residual learning structure across different networks.

## 6.4 Ablation Study

Table 7 illustrates the effectiveness of each component in PRNet. **SCE** adopts a single SCE Encoder to encode the closeness, periodic closeness, and periodic predictions, which is equivalent to the PRNet without the periodic residual learning mechanism. **SCE w/o PC** only adopts a single SCE Encoder to encode the closeness and periodic predictions. **SCE w/o S** is SCE model without the Spatial Enhance Module (SEM). **w/o R** is PRNet without the residual learning module, which utilizes seven shared parameters SCE Encoders to encode seven observed segments. **w abs** uses PRNet to predict the absolute values rather than residuals.

**Table 7: Ablation studies of PRNet on TaxiBJ-P4.**

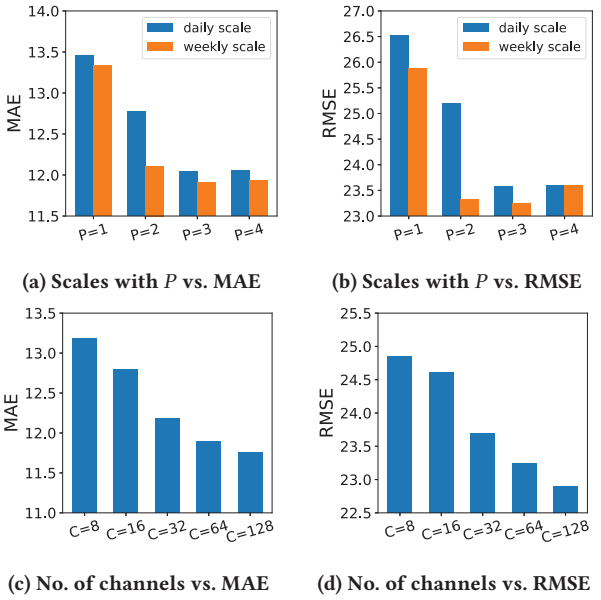
| Method | # Params | MAE | RMSE |
|---|---|---|---|
| SCE | 712K | 12.12 ± 0.11 | 23.78 ± 0.15 |
| SCE w/o PC | 707K | 12.29 ± 0.16 | 24.34 ± 0.30 |
| SCE w/o S | 703K | 12.39 ± 0.29 | 24.65 ± 0.97 |
| w/o R | 703K | 36.61 ± 0.41 | 62.20 ± 0.47 |
| w abs | 711K | 12.31 ± 0.07 | 24.35 ± 0.35 |
| PRNet (Ours) | 711K | **11.90 ± 0.05** | **23.25 ± 0.13** |

According to the results shown in Table 7, we can observe that: 1) Periodic closeness is important in prediction tasks. The reason is that it can provide the reference for time-series shifting between periodic predictions and future conditions. 2) Residual learning is essential for our model. Without it, the model cannot capture the correlations between the multi-scale time intervals because it only encodes the historical observations with seven shared parameters SCE Encoders separately. Differing from **SCE w/o PC** and **SCE** that model the periodic pattern implicitly by fusing all observations into one SCE Encoder, PRNet directly calculates dependencies of multi-scale time intervals, which provides an elegant solution for explicit periodicity representation without introducing redundant parameters. 3) Enhancing spatial information boosts model performance. Our SEM provides the most salient features based on city-scale grids, which further promotes model performance. 4) Predicting the residual instead of absolute values leads to noticeable improvement, which proves our assumption about learning residual is much easier. In summary, the experimental results and parameter comparison show that PRNet successfully captures the periodicity information as well as complex spatio-temporal correlations without increasing the model complexity.

## 6.5 Effects of Hyperparameters

In Fig. 7, we study the effects of hyperparameters in PRNet over TaxiBJ-P4. First, we study the effects of different periodic scales with different number of selected periods $P$ in Fig. 7(a)-(b). Specifically, we explore different $P$ values from 1 to 4 under the daily scale and the weekly scale. The length of period $l$ is set to one day for daily scale and one week for weekly scale. From the results, we can observe that: 1) The network with the weekly scale consistently outperforms its daily scale counterpart. The reason is that weekly differencing can provide more stable residuals as the weekly crowd flow pattern tends to be similar, while the crowd flow pattern of two successive days can be different (e.g., Friday evening and Thursday

evening). 2) Increasing the number of $P$ can improve the model performance. Because by considering multiple periodic segments, the network is allowed to attend to the information from different residual references. Then the network can perform more robustly even if the sudden variation happens between two periodic segments. However, using a very large $P$ value (i.e., $P = 4$) also degrades the performance as the training samples are reduced. We achieve the lowest MAE and RMSE when the $P$ value is 3, therefore, we choose $P = 3$ as our default setting. We also study the effects of the number of channels $C$ by attempting different values of $C$ (i.e., 8, 16, 32, 64, and 128) in Fig. 7(c)-(d). According to the results, increasing $C$ from 8 to 128 can reduce the MAE and RMSE because the model capability is improved. However, the network with 128 channels has 2802K parameters, indicating it needs 3.94 times more parameters than the network with 64 channels. As our model achieves good performance when $C$ is 64, we use $C = 64$ as the default setting.



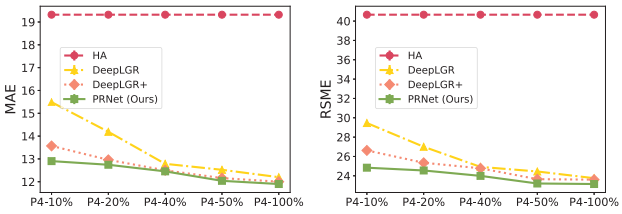

(a) Scales with $P$ vs. MAE (b) Scales with $P$ vs. RMSE

(c) No. of channels vs. MAE (d) No. of channels vs. RMSE

**Figure 7: Effects of hyperparameters on TaxiBJ-P4.**

## 6.6 Study on Training Data Budget

In real-world applications, the available data budget for network training may varied. Thus, we investigate the performance of our proposed network under different sizes of training data budgets on TaxiBJ-P4 in Fig. 8. From the results, we can observe that the models with our proposed structure, i.e., DeepLGR+ and PRNet, surpass both HA and DeepLGR given various sizes of training data budgets. Specifically, they outperform DeepLGR by a large margin given a small size of training data (10% ratio data budget). Because our proposed structure explicitly captures the periodic residual which works as a strong periodic prior that provides the statistical knowledge to the deep learning network. With the deep model capturing the complex ST correlations, this periodicity prior knowledge can make our model bridge the gap between the traditional method and deep method, and thus help the model to generate good results, especially with small training data budgets.

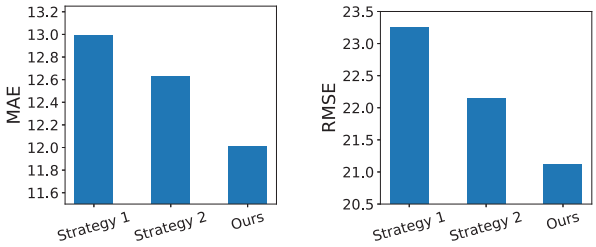



(a) Result on MAE (b) Result on RMSE

**Figure 8: Prediction results of PRNet under various data budgets. The training data are sampled from the original dataset with different ratios, i.e., 10%, 20%, 40%, 50%, and 100%.**

## 6.7 Study on Strategies for Missing Data

Fig. 9 shows the effectiveness of our proposed missing data strategy. We explore three strategies discussed in Section 6.1.1 on TaxiBJ-P4, which has a high missing ratio (i.e., 16.3%). From the results, we can observe that: 1) Strategy 1 simply excludes the samples with missing data leading to a small data budget and therefore producing inferior performance. 2) Filling the missing data with periodic default values (Ours) instead of replacing them with zero (Strategy 2) boosts the model performance. Because it introduces less extremely noisy data.




(a) Result on MAE (b) Result on RMSE

**Figure 9: Effect of missing data strategies on TaxiBJ-P2.**

## 7 CONCLUSION AND FUTURE WORK

In this paper, we studied the periodic behavior in crowd flow and proposed PRNet, a deep learning architecture that integrates the statistical strategy for multi-step ahead forecasting. We further introduced a lightweight SCE Encoder to enhance the spatio-temporal representation by suppressing and refining the intermediate features. The experiments on real-world data shown the effectiveness of PRNet, which reduces the error of MAE by 5.41%~17.63% with 1.36~147.7 times fewer parameters compared with SOTA methods. Also, integrating PRNet into existing models reduced the MAE error by 5.13%~14.77% and promoted robustness by 63.64%~80.25%. It demonstrated the potential of bridging the gap between the traditional time-series approaches and deep neural networks. This work highlights the inadequacy of previous works on periodicity modeling and sheds some light on exploiting traditional statistics to boost the deep learning model performance. Moreover, PRNet is not limited to crowd flow forecasting. In the future, we will evaluate it on other tasks that contain strong periodicity.

## 8 ACKNOWLEDGEMENT

This research is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

## REFERENCES

[1] Mohammed S Ahmed and Allen R Cook. 1979. *Analysis of freeway traffic time-series data by using Box-Jenkins techniques.* Number 722.

[2] Peter J Brockwell, Peter J Brockwell, Richard A Davis, and Richard A Davis. 2016. *Introduction to time series and forecasting.* Springer.

[3] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. 2019. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3146–3154.

[4] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proceedings of the AAAI conference on artificial intelligence*. 3656–3663.

[5] Liangzhe Han, Bowen Du, Leilei Sun, Yanjie Fu, Yisheng Lv, and Hui Xiong. 2021. Dynamic and Multi-faceted Spatio-temporal Deep Learning for Traffic Speed Forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 547–555.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[7] Minh X Hoang, Yu Zheng, and Ambuj K Singh. 2016. FCCF: forecasting citywide crowd flows based on big data. In *Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems*. 1–10.

[8] Qibin Hou, Daquan Zhou, and Jiashi Feng. 2021. Coordinate attention for efficient mobile network design. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13713–13722.

[9] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

[10] Beijing Transport Institute. 2021. Beijing Transport Development Annual Report 2021. https://www.bjtrc.org.cn/Show/download/id/68/at/0.html

[11] Guangyin Jin, Huan Yan, Fuxian Li, Yong Li, and Jincai Huang. 2021. Hierarchical Neural Architecture Search for Travel Time Estimation. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*. 91–94.

[12] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

[13] Ting Li, Junbo Zhang, Kainan Bao, Yuxuan Liang, Yexin Li, and Yu Zheng. 2020. Autost: Efficient neural architecture search for spatio-temporal prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 794–802.

[14] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu. 2018. Multi-task representation learning for travel time estimation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1695–1704.

[15] Yuxuan Liang, Kun Ouyang, Lin Jing, Sijie Ruan, Ye Liu, Junbo Zhang, David S. Rosenblum, and Yu Zheng. 2019. UrbanFM: Inferring Fine-Grained Urban Flows. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '19)*. ACM, 3132–3142.

[16] Yuxuan Liang, Kun Ouyang, Junkai Sun, Yiwei Wang, Junbo Zhang, Yu Zheng, David Rosenblum, and Roger Zimmermann. 2021. Fine-Grained Urban Flow Prediction. In *Proceedings of the Web Conference 2021*. 1833–1845.

[17] Yuxuan Liang, Kun Ouyang, Yiwei Wang, Ye Liu, Junbo Zhang, Yu Zheng, and David S. Rosenblum. 2020. Revisiting Convolutional Neural Networks for City-wide Crowd Flow Analytics. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020, Proceedings, Part I*, Vol. 12457. Springer, 578–594.

[18] Ziqian Lin, Jie Feng, Ziyang Lu, Yong Li, and Depeng Jin. 2019. Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. In *Proceedings of the AAAI conference on artificial intelligence*. 1020–1027.

[19] Kun Ouyang, Yuxuan Liang, Ye Liu, Zekun Tong, Sijie Ruan, David Rosenblum, and Yu Zheng. 2020. Fine-grained urban flow inference. *IEEE transactions on knowledge and data engineering* (2020).

[20] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. 2018. BAM: Bottleneck Attention Module. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. BMVA Press, 147.

[21] Shashi Shekhar, Zhe Jiang, Reem Y Ali, Emre Eftelioglu, Xun Tang, Venkata MV Gunturi, and Xun Zhou. 2015. Spatiotemporal data mining: A computational perspective. *ISPRS International Journal of Geo-Information* 4, 4 (2015), 2306–2338.

[22] Hongzhi Shi and Yong Li. 2018. Discovering periodic patterns for large scale mobile traffic data: Method and applications. *IEEE Transactions on Mobile Computing* 17, 10 (2018), 2266–2278.

[23] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* 28 (2015).

[24] Junkai Sun, Junbo Zhang, Qiaofei Li, Xiuwen Yi, Yuxuan Liang, and Yu Zheng. 2020. Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[25] Quang Thanh Tran, Zhihua Ma, Hengchao Li, Li Hao, and Quang Khai Trinh. 2015. A multiplicative seasonal ARIMA/GARCH model in EVN traffic prediction. *International Journal of Communications, Network and System Sciences* 8, 4 (2015), 43.

[26] U.S. Department OF Transportation. 2020. Transportation Statistics Annual Report 2020. https://rosap.ntl.bts.gov/view/dot/53936/dot_53936_DS1.pdf?download-document-submit=Download

[27] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. In *The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016*. ISCA, 125.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations* (2018).

[30] Billy M Williams. 1999. *Modeling and forecasting vehicular traffic flow as a seasonal stochastic time series process.* University of Virginia.

[31] Billy M Williams and Lester A Hoel. 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of transportation engineering* 129, 6 (2003), 664–672.

[32] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*. 3–19.

[33] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. ijcai.org, 1907–1913.

[34] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. 2020. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908* (2020).

[35] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. 2019. Revisiting Spatial-Temporal Similarity: A Deep Learning Framework for Traffic Prediction. In *2019 AAAI Conference on Artificial Intelligence (AAAI'19)*.

[36] Quan Yuan, Wei Zhang, Chao Zhang, Xinhe Geng, Gao Cong, and Jiawei Han. 2017. PRED: Periodic region detection for mobility modeling of social media users. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 263–272.

[37] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-first AAAI conference on artificial intelligence*.

[38] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. 2016. DNN-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems*. 1–4.

[39] Xiyue Zhang, Chao Huang, Yong Xu, Lianghao Xia, Peng Dai, Liefeng Bo, Junbo Zhang, and Yu Zheng. 2021. Traffic Flow Forecasting with Spatial-Temporal Graph Diffusion Network. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*. AAAI Press, 15008–15015.

[40] Yingxue Zhang, Yanhua Li, Xun Zhou, Jun Luo, and Zhi-Li Zhang. 2022. Urban traffic dynamics prediction—a continuous spatial-temporal meta-learning approach. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 2 (2022), 1–19.

[41] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 1234–1241.

[42] Fan Zhou, Liang Li, Kunpeng Zhang, and Goce Trajcevski. 2021. Urban flow prediction with spatial–temporal neural ODEs. *Transportation Research Part C: Emerging Technologies* 124 (2021), 102912.

[43] Ali Zonoozi, Jung-jae Kim, Xiao-Li Li, and Gao Cong. 2018. Periodic-CRN: A Convolutional Recurrent Model for Crowd Density Prediction with Recurring Periodic Patterns.. In *IJCAI*. 3732–3738.