International Journal of Digital Earth

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tjde20

Lightweighting the prediction process of urban states with parameter sharing and dilated operations

Peixiao Wang, Haolong Yang, Hengcai Zhang, Shifen Cheng, Feng Lu & Zeqiang Chen

To cite this article: Peixiao Wang, Haolong Yang, Hengcai Zhang, Shifen Cheng, Feng Lu & Zeqiang Chen (2025) Lightweighting the prediction process of urban states with parameter sharing and dilated operations, International Journal of Digital Earth, 18:1, 2468414, DOI: 10.1080/17538947.2025.2468414

To link to this article: https://doi.org/10.1080/17538947.2025.2468414

INTERNATIONAL JOURNAL OF DIGITAL EARTH
2025, VOL. 18, NO. 1, 2468414
https://doi.org/10.1080/17538947.2025.2468414

Lightweighting the prediction process of urban states with parameter sharing and dilated operations

Peixiao Wang [a,b], Haolong Yang [c], Hengcai Zhang [a,b], Shifen Cheng [a,b], Feng Lu [a,b] and Zeqiang Chen [d]

[a]State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, CAS, Beijing, People's Republic of China; [b]College of Resources and Environment, University of Chinese Academy of Sciences, Beijing, People's Republic of China; [c]Gina Cody School of Engineering and Computer Science, Concordia University, Montreal, Canada; [d]National Engineering Research Center for Geographic Information System, China University of Geosciences, Wuhan, People's Republic of China

ABSTRACT

Lightweight and high-precision prediction models for urban states are anticipated to run efficiently on resource-limited devices, serving as key technologies for realizing smart city management. However, many existing models, despite achieving high prediction precision, suffer from overly complex designs, leading to low computational efficiency, a large number of learnable parameters, and difficulty in hyper-parameter calibration. In this study, we present a lightweight parameter-shared dilated convolutional network (PSDCN) to address these challenges. Specifically, we define parameter-shared temporal/graph dilated convolution operators to efficiently and accurately capture spatio-temporal correlations without significantly increasing model's computation time and scale of learnable parameters. Furthermore, we establish mathematical relationships between hyperparameters, significantly reducing their number and simplifying the calibration process. The PSDCN model was validated using $PM_{2.5}$, traffic, and temperature datasets. The results demonstrated that the PSDCN model simplifies hyperparameter calibration. It also either outperforms or matches the prediction accuracy of nine baselines, while achieving better time efficiency and requiring fewer learnable parameters.

ARTICLE HISTORY
Received 28 October 2024
Accepted 7 February 2025

KEYWORDS
Urban states; spatio-temporal prediction; dilated operation; parameter sharing; hyper-parameter dependence

1. Introduction

The rapid development of the Internet of Things (IoT) has enabled real-time monitoring of urban systems, providing an essential data source for predicting urban states (Zhou et al. 2022; W. Zhang et al. 2025). Currently, prediction technologies are increasingly used to support urban traffic management and public health protection (Karl et al. 2024; Wang et al. 2024). For instance, accurately predicting future traffic states can balance network traffic flow, alleviating urban congestion (Guangyue Li et al. 2024; Y. Xu et al. 2023). Similarly, accurately predicting future air quality can protect humans from exposure to heavily polluted environments by issuing early warning signals (L. Mengfan et al. 2022; Zhang et al. 2021).

CONTACT Hengcai Zhang zhanghc@lreis.ac.cn State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, CAS, Beijing 100101, People's Republic of China; College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China
Supplemental data for this article can be accessed online at https://doi.org/10.1080/17538947.2025.2468414.

Prediction technologies for urban states are a type of spatio-temporal prediction models, typically classified into three main categories: statistical learning, machine learning, and deep learning models(Cheng et al. 2024; Xie et al. 2020). Among these, deep learning models are the dominant prediction technologies due to their ability to capture complex dependencies and patterns (H. Wang et al. 2023; Y. Zhang et al. 2023; Q. Zheng et al. 2023). At present, related scholars developed a variety of deep learning models aimed at enhancing prediction precision (T. Zhang, Liu, and Wang 2022; T. Zhang, Liu, et al. 2024; T. Zhang, Wang, et al. 2024). From the early recurrent neural networks (Chung et al. 2014) to the latest spatio-temporal graph neural networks (Y. Liu et al. 2024), the prediction precision of these models has steadily advanced. However, as prediction precision improves, the increasing complexity of current models presents challenges for their practical application in real-world scenarios (Do et al. 2019; Guanyao Li et al. 2023). First, overly complex models may be computationally inefficient, limiting their deployment on real-time applications (P. Wang, Zhang, Cheng, et al. 2024), such as scenarios requiring rapid prediction of traffic flow or air quality. Second, these models may have a large number of learnable parameters, restricting their deployment on resource-limited devices, such as mobile or edge computing devices. Finally, these models may contain a large number of hyper-parameters, making convergence more challenging (Yang and Shami 2020). Even if the model does converge, it may only reach a local optimum rather than a global one.

In real-world scenarios, it is essential to develop lightweight and highly precision prediction models. However, many existing models still face challenges in balancing prediction precision with ease of use. Therefore, we present a new Parameter-Shared Dilated Convolutional Network (PSDCN) to address these challenges, with key contributions including:

(1) Inspired by parameter sharing and dilation computation, we define the lightweight Parameter-Shared Temporal Dilated Convolution (PSTDC) operator and Parameter-Shared Graph Dilated Convolution (PSGDC) operator. The PSTDC and PSGDC operators can efficiently capture spatio-temporal correlations without significantly increasing the computation time and the scale of learnable parameters.
(2) We establish logical relationships among hyperparameters to simplify both the number and complexity of hyperparameter tuning, thereby reducing the time and resources required for trial-and-error during model calibration.
(3) Three datasets ($PM_{2.5}$, traffic, and temperature datasets) were utilized to assess model prediction performance, including prediction precision, computational efficiency, and scale of learnable parameters. The results demonstrate that the proposed PSDCN model is well suited to be deployed on resource-limited devices for real-time demanding prediction tasks, such as rapid urban air quality prediction on mobile phones.

2. Related works

Given the advantages of deep learning models in prediction tasks, we primarily focus on deep learning-based spatio-temporal prediction models. In this subsection, we first examine the base operators that constitute spatio-temporal prediction models, followed by a review of more complex models built upon these operators. Additionally, we review existing lightweight spatio-temporal prediction models relevant to this study.

2.1. Basic neural network operators

Currently, most spatio-temporal prediction models are hybrid deep learning models (G. Zheng et al. 2023). In this study, we define the smallest unit that constitutes a hybrid deep learning model as the basic neural network operator. These basic neural network operators can be broadly categorized into two main groups: those that capture temporal dependencies and those that capture spatial dependencies (Wang et al. 2023; Xie et al. 2020). Common neural network operators for

capturing temporal dependencies include recurrent neural network (RNN) operator (Chung et al. 2014), one-dimensional convolutional (1D-CNN) operator (Yu, Yin, and Zhu 2018), temporal attention (TAtten) operator (Tan et al. 2023), and neural ordinary differential equation (NODE) operator (R. T. Q. Chen et al. 2018). Common neural network operators for capturing spatial dependencies include two-dimensional convolutional (2D-CNN) operator (Q. Li, Wang, and Li 2021), graph convolutional operator (GCN) (Kipf and Welling 2017), and spatial attention (SAtten) operator (M. Xu et al. 2021). Basic operators form the foundation for developing complex spatio-temporal prediction models. By combining or improving these operators, spatio-temporal prediction models can be customized to address the specific needs of various scenarios (Wang et al. 2023).

2.2. Spatio-temporal prediction models based on hybrid deep learning

Complex spatio-temporal prediction models improve prediction precision by fusing basic neural network operators, can be categorized into grid-based prediction models, graph-based prediction models and transformer-based prediction models.

Grid-based prediction models are an early class of prediction models, typically combining 2D-CNN operators with RNN or 1D-CNN operators, such as the ConvGRU model (Shi et al. 2017) and ST-ResNet model (Jia and Yan 2021). While these models are computationally efficient, they are limited by their insufficient prediction precision.

Graph-based prediction models have emerged as a high-precision prediction approach in recent years, typically combining GCN operators with RNN, 1D-CNN, or NODE operators. Notable examples include the T-GCN model (Zhao et al. 2020), the ASTGCN model (Guo et al. 2019), the DSTAGNN model (Lan et al. 2022), the BiSTGN model (Wang et al. 2022), the GDGCN model (Y. Xu et al. 2023), the STGODE model (Fang et al. 2021), and the STA-ODE model (. Wang, Zhang, Zhang, et al. 2024). Additionally, some scholars have integrated weather factor and Points-of-Interest factor to enhance model prediction precision, such as STECA-GCN model (S. Liu et al. 2023) and the MB-TGCN model (Guan et al. 2024). At present, existing graph-based prediction models have achieved notable improvements in prediction precision. However, as prediction precision improves, the design of graph-based models becomes increasingly complex, limiting their application in real-world scenarios (Do et al. 2019; Guanyao Li et al. 2023).

Transformer-based prediction models are also a high-precision approach, typically combining SAtten and TAtten operators, such as the STTNs model (M. Xu et al. 2021), the AirFormer model (Liang et al. 2023), and the TAU model (Tan et al. 2023). Similar to graph-based prediction models, transformer-based prediction models also suffer from the issue of overly complex design. This issue arises from the fact that, from a mathematical standpoint, transformer-based and graph-based prediction models can be unified and share comparable levels of complexity.

2.3. Lightweight spatio-temporal prediction models

Lightweight models achieve satisfactory prediction precision without significantly increasing model complexity, especially for graph-based and transformer-based prediction models (Wang et al. 2025). In recent years, some scholars have explored model lightweighting and developed several lightweight spatio-temporal prediction models. For example, Chien and Huang (2021) proposed a lightweight LSTCNN model to reduce the model's complexity through compression techniques such as pruning, quantization, co-training, and feature extraction. Li et al. (2023) proposed a lightweight ST-TIS model to decrease the model computational complexity from $O(n^2)$ to $O(n\sqrt{n})$ using region connectivity graphs. Wang et al. (2024) proposed a lightweight ST-GDN model to improve the model's computational efficiency by reducing the depth of neural networks. Compared to models without lightweight design, existing lightweight models are still relatively limited. Additionally, these models still face challenges such as the large scale of learnable parameters and the complexity of hyper-parameter tuning (Cheng, Peng, and Lu 2020; Guanyao Li et al. 2023).

2.4. Challenges and strategies

In general, most existing prediction models focus on improving prediction precision while neglecting the importance of model lightweighting for ease of use. Although a few lightweight models have been proposed, they mainly focus on improving computational efficiency, neglecting issues related to the scale of learnable parameters and the complexity of hyperparameter tuning.

To address the above challenges, we present a new lightweight PSDCN model for spatio-temporal prediction. First, while ensuring prediction precision, the parameter sharing mechanism is employed to reduce the scale of learnable parameters, and the dilated operation is utilized to enhance the model's computational efficiency. Second, the mathematical relationships among hyper-parameters are established to reduce the difficulty of hyper-parameter tuning.

3. Preliminaries

In this study, the proposed PSDCN model is a graph-based prediction model. Before delving into its details, we first provide the relevant definitions. In the graph $G = (V, \boldsymbol{A})$, each sensor can be represented as a node $v_i \in V$, and the relationship between sensors v_i and v_j can be abstracted as an edge $\boldsymbol{A}_{ij} \in \boldsymbol{A}$. The spatio-temporal data collected by sensor v_i in tth time window can be represented as x_i^t. Generally, the spatio-temporal data collected by all sensors across all time windows form a matrix $\boldsymbol{X} \in \mathcal{R}^{n \times T}$, where n denotes the total count of sensors and T denotes the total count of time windows.

The aim of this study is to establish a lightweight model, as explicitly formulated in Equation (1).

$$(\hat{\boldsymbol{x}}^{T+q}, \ldots, \hat{\boldsymbol{x}}^{T+2}, \hat{\boldsymbol{x}}^{T+1}) = \mathcal{M}_{\text{PSDCN}} \leftarrow (\boldsymbol{x}^{T}, \boldsymbol{x}^{T-1}, \ldots, \boldsymbol{x}^{T-p+1} | G, \boldsymbol{W}) \tag{1}$$

where $\mathcal{M}_{\text{PSDCN}}$ denotes the PSDCN model; $\boldsymbol{x}^{T-p+1} = \{x_i^{T-p+1}\}_{i=1}^{n} \in R^{n \times 1}$ denotes the historical spatio-temporal data, with p being the historical dependency horizon; $\hat{\boldsymbol{x}}^{T+q} = \{\hat{x}_i^{T+q}\}_{i=1}^{n} \in R^{n \times 1}$ denotes the predicted spatio-temporal data, with q being the prediction horizon; $\boldsymbol{W}$ denotes the learnable parameters in the PSDCN model.

4. Proposed approach

The structure of the PSDCN model is depicted in Figure 1: the PSDCN model is composed of multiple parameter-shared modules, each incorporating L spatio-temporal blocks and a hyper-parameter dependency component. During model forward propagation, L spatio-temporal blocks share the same set of convolutional kernels and efficiently capture complex nonlinear dependencies in spatio-temporal data. Specifically, the PSGDC operator within the spatio-temporal block is utilized to efficiently capture spatial dependencies, while the PSTDC operator within the spatio-temporal block is utilized to efficiently capture temporal dependencies. Additionally, the hyper-parameter dependency component automatically determines the optimal number of spatio-temporal

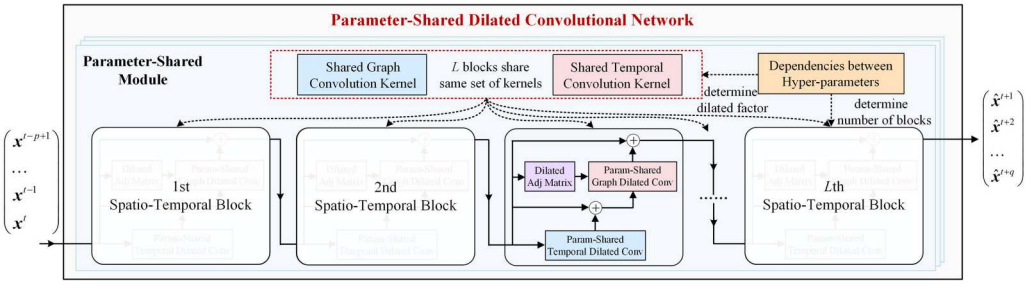

Figure 1. Structure of the PSDCN model.

blocks and the optimal size of the dilated factor via the calibrated kernel size and calibrated historical dependency horizon. By reducing the calibration number of hyper-parameter, we mitigate the complexity inherent in model tuning.

4.1. Construction of the PSDCN

In this subsection, we offer a comprehensive explanation of the forward propagation for the PSDCN model. Although the PSDCN model consists of multiple parameter-shared modules, typically only one parameter-shared module is needed to address most simple spatio-temporal prediction tasks due to hyper-parameter dependency component (discussed in Section 4.1.3).

When there is only one parameter-shared module, the input data will sequentially pass through convolutional layers, L spatio-temporal blocks, and another convolutional layer to obtain the final output. Specifically, the spatio-temporal data $\{\boldsymbol{x}^t\}_{t-p+1}^{t} \in R^{n\times p}$ first undergo a convolutional layer to increase the data dimensionality, obtaining the input tensor $\mathcal{H}_1 \in R^{n\times p\times f}$ for the first spatio-temporal block. Then, input tensor $\mathcal{H}_1$ sequentially passes through L spatio-temporal blocks to get output tensor $\mathcal{H}_{L+1} \in R^{n\times p\times f}$ for the last spatio-temporal block. Finally, output tensor $\mathcal{H}_{L+1}$ undergoes dimensionality reduction through another convolutional layer to obtain the final prediction $\{\hat{\boldsymbol{x}}\}_{t+1}^{t+q} \in R^{n\times q}$. Equations (2) and (3) further show the forward propagation of the PSDCN model mathematically.

$$\text{PSDCN}(\{\boldsymbol{x}^t\}_{t-p+1}^{t}):\begin{cases}\mathcal{H}_2 = STBlock(Conv(\{\boldsymbol{x}^t\}_{t-p+1}^{t})|\boldsymbol{W}_T, \boldsymbol{W}_G) & l=1\\ \mathcal{H}_{l+1} = STBlock(\mathcal{H}_l|\boldsymbol{W}_T, \boldsymbol{W}_G) & 1<l<L\\ \{\hat{\boldsymbol{x}}\}_{t+1}^{t+q} = Conv(STBlock(\mathcal{H}_L|\boldsymbol{W}_T, \boldsymbol{W}_G)) & l=L\end{cases} \tag{2}$$

$$\text{STBlock}(\mathcal{H}_l|\boldsymbol{W}_T, \boldsymbol{W}_G):\begin{cases}\dot{\dot{\mathcal{H}}}_l\, PSTDC(\mathcal{H}_l|\boldsymbol{W}_T)\\ \ddot{\ddot{\mathcal{H}}}_l\, Relu\left(\dot{\mathcal{H}}_l\right) + \mathcal{H}_l\\ \dddot{\mathcal{H}}_l = PSGDC\left(\ddot{\mathcal{H}}_l\, \boldsymbol{W}_G\right)\\ \mathcal{H}_{l+1} = Norm\left(Relu\left(\dddot{\mathcal{H}}_l\right) + \mathcal{H}_l\right)\end{cases} \tag{3}$$

where $\{\boldsymbol{x}^t\}_{t-p+1}^{t} \in R^{n\times p}$ denotes the input of the PSDCN model, with p being the historical dependency horizon; $\{\hat{\boldsymbol{x}}\}_{t+1}^{t+q} \in R^{n\times q}$ represents the output of the PSDCN model, with q being the prediction horizon; *STBlock* is the spatio-temporal block in the parameter-shared module; $\mathcal{H}_l \in R^{n\times p\times f}$ and $\mathcal{H}_{l+1} \in R^{n\times p\times f}$ represent the input and output tensors of the l-th spatio-temporal block, with f being the dimension of the hidden layer; $\boldsymbol{W}_T$ and $\boldsymbol{W}_G$ are the learnable parameters shared within L spatio-temporal blocks; *PSTDC* and *PSGDC* represent the operators for mining temporal dependencies and spatial dependencies, respectively; $\dot{\dot{\mathcal{H}}}_l \in R^{n\times p\times f}$, $\ddot{\ddot{\mathcal{H}}}_l \in R^{n\times p\times f}$, and $\dddot{\mathcal{H}}_l \in R^{n\times p\times f}$ are temporary variables in PSTDC and PSGDC operators; *Conv* represents the convolution function for dimension alignment; *Relu* represents the activation function; *Norm* represents the parameter regularization function. Equations (2) and (3) highlight that the essence of the proposed PSDCN model lies in two key aspects: *Definition of the PSTDC operator* and *Definition of the PSGDC operator*.

4.1.1. Parameter-Shared temporal dilated convolutional operator

Aiming to address the issues of over-complex design, low computational efficiency, and large scales of learnable parameters, we extend the classical temporal convolution operator to efficiently mine temporal dependencies in data. Illustrated with a two-layer temporal convolutional network as an example, Figure 2 demonstrates the extension concept of the classical temporal convolution operator. When the kernel size is set to 3, a two-layer network can model only 5 time windows using the

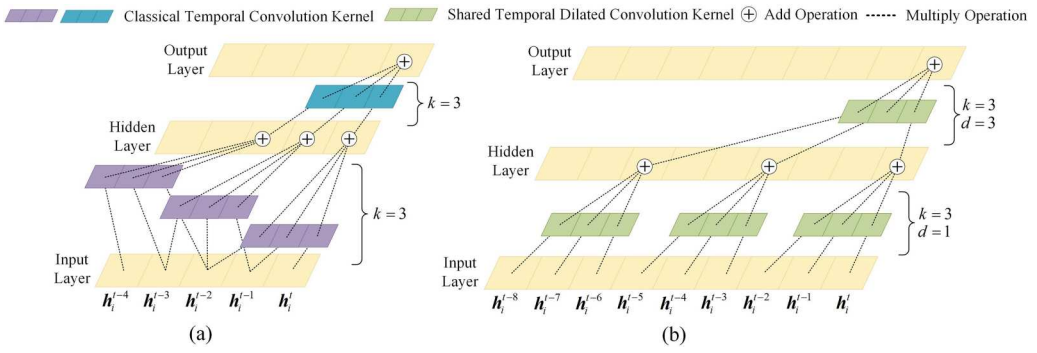

Figure 2. Extension concept of the classical temporal convolution: (a) classical temporal convolution, and (b) parameter-shared temporal dilated convolution.

classical temporal convolution operator, whereas it can model 9 time windows using the temporal dilated convolution operator. In other words, with the historical dependency horizon fixed, the temporal dilated convolution operator can capture more temporal dependencies with fewer layers, thus improving the computational efficiency of forward propagation. Furthermore, we find that introducing the dilation factor does not change the kernel size of the convolution but only its calculation method. This finding implies that we can share kernels between different layers, reducing the scale of learnable parameters in the optimization process. Inspired by these observations, we propose the lightweight PSTDC operator.

In contrast to the classical temporal convolutional operator, the PSTDC operator presents two primary advantages. First, the PSTDC operator regulates the network depth via the dilation factor, improving forward propagation efficiency. Second, the PSTDC operator facilitates kernel sharing across layers, diminishing the scale of learnable parameters during optimization. Taking the tensor $\mathcal{H}_l$ as an example, Equation (4) demonstrates the computation of the PSTDC operator.

$$PSTDC(\mathcal{H}_l|\boldsymbol{W}_T) = \begin{pmatrix} \sum_{k=1}^{K} \boldsymbol{w}_T^k \odot \boldsymbol{h}_1^{l:(t-p+1)-(K-k)d_l^T} & \cdots & \sum_{k=1}^{K} \boldsymbol{w}_T^k \odot \boldsymbol{h}_1^{l:t-(K-k)d_l^T} \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^{K} \boldsymbol{w}_T^k \odot \boldsymbol{h}_n^{l:(t-p+1)-(K-k)d_l^T} & \cdots & \sum_{k=1}^{K} \boldsymbol{w}_T^k \odot \boldsymbol{h}_n^{l:t-(K-k)d_l^T} \end{pmatrix} \tag{4}$$

where $\mathcal{H}_l = \{\{\boldsymbol{h}_i^{l:t-\tau+1}\}_{i=1}^n\}_{\tau=1}^p \in R^{n\times p\times f}$ represents the input tensor of the PSTDC operator in the l-th spatio-temporal block, with $\boldsymbol{h}_i^{l:t} \in R^{f\times 1}$ being the state of node v_i in the t-th time window; $\boldsymbol{W}_T = \{\boldsymbol{w}_T^k\}_{k=1}^K \in R^{f\times K}$ represents the shared temporal kernels, with K being the kernel size and $\boldsymbol{w}_T^k \in R^{f\times 1}$ being the k-th shared kernel; d_l^T represents the dilation factor of the PSTDC operator in the l-th spatio-temporal block; and $\odot$ represents the vector product. **Note:** We need to input the kernel $\boldsymbol{W}_T$ into the next spatio-temporal block to enable parameter sharing in the l-th and $(l+1)$th spatio-temporal blocks.

4.1.2. Parameter-Shared graph dilated convolutional operator

After applying the PSTDC operator in the l-th spatio-temporal block, we obtain the tensor $\mathcal{H}_l \ddot{\in} R^{n\times p\times f}$. Similar to temporal dimension, we extend the classical graph convolutional operator to efficiently explore spatial dependencies in tensor $\mathcal{H}_l$. Illustrated with a two-layer graph convolutional network as an example, Figure 3 demonstrates the extension concept of the classical graph convolution operator. For a target node, a two-layer network using the classical graph convolution operator can model only 6 spatial neighbors, whereas it can model 9 spatial neighbors using the graph dilated convolution operator. This indicates that introducing the dilated factor can capture

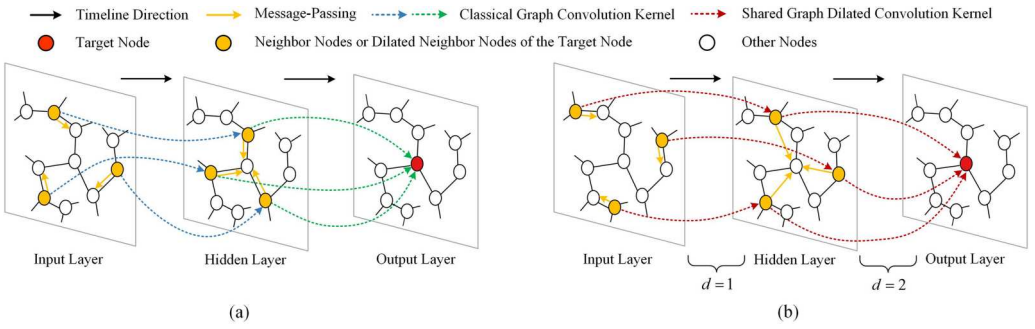

Figure 3. Extension concept of the classical graph convolution: (a) classical graph convolution, and (b) parameter-shared graph dilated convolution.

more distant spatial dependencies with fewer layers, enhancing the computational efficiency of forward propagation. Additionally, the dilation factor does not change the kernel size of most classical graph convolution operators (see Appendix A), meaning we can share graph kernels between different layers to reduce the scale of learnable parameters in the optimization stage. Inspired by these observations, we propose the lightweight PSGDC operator.

In the spatial dimension, the PSGDC operator also has two notable advantages compared to the classical graph convolutional operator. First, it improves the computational efficiency of forward propagation, and second, it reduces the scale of learnable parameters in the optimization process. Additionally, to enhance the prediction capability of the PSGDC operator, we design its forward propagation based on graph attention, as shown in Equations (5) and (6).

$$PSGDC\left(\ddot{\mathcal{H}}_l \boldsymbol{W}_S\right) = \begin{pmatrix} \sum_{j\in \boldsymbol{A}_1^{d_l^S}} \gamma_{j1}^{l:(t-p+1)} \ddot{\boldsymbol{h}}\, 1^{l:(t-p+1)} \boldsymbol{W}_S^V & \cdots & \sum_{j\in \boldsymbol{A}_1^{d_l^S}} \gamma_{j1}^{l:t} \ddot{\boldsymbol{h}}\, 1^{l:t} \boldsymbol{W}_S^V \\ \vdots & \ddots & \vdots \\ \sum_{j\in \boldsymbol{A}_n^{d_l^S}} \gamma_{jn}^{l:(t-p+1)} \ddot{\boldsymbol{h}}\, n^{l:(t-p+1)} \boldsymbol{W}_S^V & \cdots & \sum_{j\in \boldsymbol{A}_n^{d_l^S}} \gamma_{jn}^{l:t} \ddot{\boldsymbol{h}}\, 1^{l:t} \boldsymbol{W}_S^V \end{pmatrix} \tag{5}$$

$$\gamma_{jn}^{l:t} = \frac{exp\left(Relu\left(\left[\ddot{\boldsymbol{h}}\, j^{l:t} \middle| \ddot{\boldsymbol{h}}\, n^{l:t}\right] \boldsymbol{W}_S^Q\right)\right)}{\sum_{k\in \boldsymbol{A}_n^{d_l^S}} exp\left(Relu\left(\left[\ddot{\boldsymbol{h}}\, k^{l:t} \middle| \ddot{\boldsymbol{h}}\, n^{l:t}\right] \boldsymbol{W}_S^Q\right)\right)} \tag{6}$$

where $\dddot{\mathcal{H}}_l \left\{ \left\{ \ddot{\boldsymbol{h}}\, i^{l:t-\tau+1} \right\}_{i=1}^{n} \right\}_{\tau=1}^{p} \in R^{n\times p\times f}$ denotes the input tensor of the PSGDC operator in the l-th spatio-temporal block, with $\ddot{\boldsymbol{h}}\, i^{l:t} \in R^{f\times 1}$ being the state of node v_i in the t-th time window; γ_{jn} denotes the impact weight of node v_j on node v_n; $\boldsymbol{A}_n^{d_l^S}$ represents the dilation factor of the PSGDC operator in the l-th spatio-temporal block, calculated using the method provided by Wang et al. (2024); $(\boldsymbol{W}_S^V, \boldsymbol{W}_S^Q) \in \boldsymbol{W}_S$ are the shared learnable parameters in the PSGDC operator; $[\cdot|\cdot]$ denotes the vector concatenation function; *Relu* represents the activation function; *exp* denotes the exponential function. **Note:** We need to input the kernel $\boldsymbol{W}_S = (\boldsymbol{W}_S^V, \boldsymbol{W}_S^Q)$ into the next spatio-temporal block to enable parameter sharing in the l-th and $(l+1)$th spatio-temporal blocks.

4.2. Optimization of the PSDCN

4.2.1. Hyper-parameter dependency component

In the construction process of the PSDCN model, we introduced several hyper-parameters, including the historical dependency horizon p, the count L of spatio-temporal blocks, the dimension f of hidden layers, the kernel size K of the temporal convolution, the dilation factor d_l^T in the PSTDC operator, and the dilation factor d_l^S in the PSGDC operator. As mentioned earlier, having too many hyper-parameters can increase the difficulty of model calibration. Therefore, we proposed a hyper-parameter dependency component to establish relationships between hyper-parameters, reducing the number of hyper-parameter calibrations.

As shown in Figure 4, when d_l^T and K exhibit a power relationship in the l-th spatio-temporal block, the PSTDC operator can fully utilize the receptive field of shared temporal convolutional kernels to capture longer time windows. Similarly, when d_l^S and 2 exhibit a power relationship, the PSGDC operator can effectively capture a broader spatial range. Furthermore, to ensure that a single parameter-shared module can handle most simple prediction tasks, we must enable the L spatio-temporal blocks to fully model p time windows. Specifically, Equations (7)—(9) define the parameter dependencies in the PSDCN model.

$$d_l^T = K^{l-1} \tag{7}$$

$$d_l^S = 2^{l-1} \tag{8}$$

$$L = \min_l (K^{l-1} > p) - 1 \tag{9}$$

where K denotes the kernel size of the shared temporal convolution; d_l^T and d_l^S represent the dilation factor in the PSTDC/PSGDC operator of the l-th spatio-temporal block; L represents the total number of spatio-temporal blocks. Once the dependencies of hyper-parameters are defined, we only need to calibrate 3 hyper-parameters instead of 6, namely p, f, and K.

4.2.2. Loss function

The PSDCN model predicts the future q spatio-temporal data from the historical p spatio-temporal data. During the optimization process, mean squared error is employed to minimize the loss between the predicted values and the actual values, as illustrated in Equation (10).

$$\mathcal{L}(\boldsymbol{W}) = \min_{\boldsymbol{W}} \sum_{j=1}^{q} \sum_{i=1}^{n} (x_i^{t+j} - \hat{x}_i^{t+j})^2 \tag{10}$$

where $x_i^{t+j} \in \mathcal{R}^{1\times 1}$ denotes the actual value of node v_i in the $(t+j)$-th time window; $\hat{x}_i^{t+j} \in \mathcal{R}^{1\times 1}$ denotes the predicted value of node v_i in the $(t+j)$-th time window

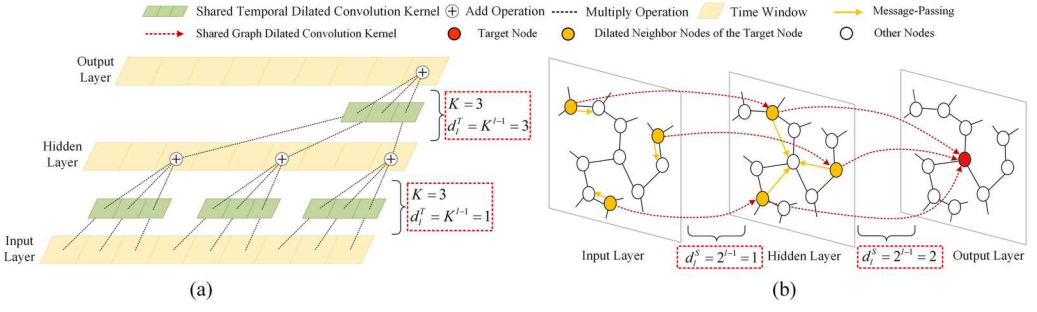

Figure 4. Illustration of hyper-parameter dependencies (a) hyper-parameters in the PSTDC operator, and (b) hyper-parameters in the PSGDC operator.

4.3. Training and optimization

This subsection details the optimization of the PSDCN model. During optimization process, historical spatio-temporal data is segmented into training and testing datasets. The training dataset is utilized to adjust the learnable parameters in the PSDCN model, while the testing dataset is employed to evaluate its prediction performance. Algorithm 1 outlines the optimization procedure of the PSDCN model. Initially, training dataset is constructed from the spatio-temporal matrix (lines 1-3). Next, we determine the number of spatio-temporal blocks using the hyper-parameter dependency component (line 5). Then, we calculate the dilation factors for each spatio-temporal block(line 9) and iteratively obtain the prediction results (line 10). Finally, the learned PSDCN model is gained by minimizing loss function (line 11).

Algorithm 1. Training Process of PSDCN
Input: Spatio-temporal matrix: $\boldsymbol{X} = \{\boldsymbol{x}^t\}_{t=1}^{T}$
Historical dependent horizon: p
Prediction horizon: q
Dimension of the hidden state: f
Kernel size: K
Output: PSDCN model: $\mathcal{M}_{PSDCN}$
1: $\Omega \leftarrow \emptyset$
2: for each $t \in [p, 2, \ldots, T-q]$:
3: construct training samples $\left\langle \{\boldsymbol{x}^{t-p+1}, \ldots\ldots, \boldsymbol{x}^{t}\}, \{\boldsymbol{x}^{t+1}, \ldots\ldots, \boldsymbol{x}^{t+q}\} \right\rangle \in \Omega$
4: initialize learnable parameters $\boldsymbol{W}$ of PSDCN
5: determine hyper-parameters L by Equation (9)
6: repeat until $\mathcal{M}_{PSDCN}$ **converges**:
7: randomly select a batch of samples Ω_b from Ω
8: for each $l \in [1, 2, \ldots, L]$:
9: determine hyper-parameters d_l^T and d_l^S by Equations (7)—(8)
10: compute tensor $\mathcal{H}_{l+1}$ or $\{\hat{\boldsymbol{x}}^{t+1}, \ldots\ldots, \hat{\boldsymbol{x}}^{t+q}\}$ by Equations (2)—(3)
11: update $\boldsymbol{W}$ by minimizing the mean squared error
12: output the trained PSDCN model $\mathcal{M}_{PSDCN}$

5. Experiments

5.1. Data sources and data preprocessing

In urban systems, real-time prediction of traffic flow, air quality, and weather is a prevalent and crucial task. Therefore, this subsection employed three publicly available datasets—$PM_{2.5}$, traffic, and temperature datasets—to assess the performance of PSDCN model (P. Wang, Zhang, Cheng, et al. 2024).

Figure 5 illustrates the spatial distribution of three spatio-temporal datasets. The $PM_{2.5}$ dataset comprises 36 monitoring sensors, operating on a 60-minute time window. The traffic dataset consists of 71 monitoring sensors, operating on a 5-minute time window. The temperature dataset encompasses 45 monitoring grids, operating on a 5-minute time window.

The data flow in this study is divided into three main steps. First, the spatio-temporal data is preprocessed to ensure it meets the input requirements of the model. Second, the forward propagation is executed to obtain the model's output. Finally, the model is optimized to obtain a prediction model suitable for practical applications. Since the forward and backward propagation processes have been described in the previous section, this section focuses on the data preprocessing procedure, as follows:

(1) Raw spatio-temporal data often contain missing values, affecting the prediction accuracy of the model. In this study, we used the BTTF model, as proposed by Chen and Sun (2022), to impute the missing values.
(2) We constructed a first-order adjacency matrix based on the similarity between spatial objects. Specifically, the first-order neighbors of the target spatial entities were identified using the 10 most similar spatial objects.

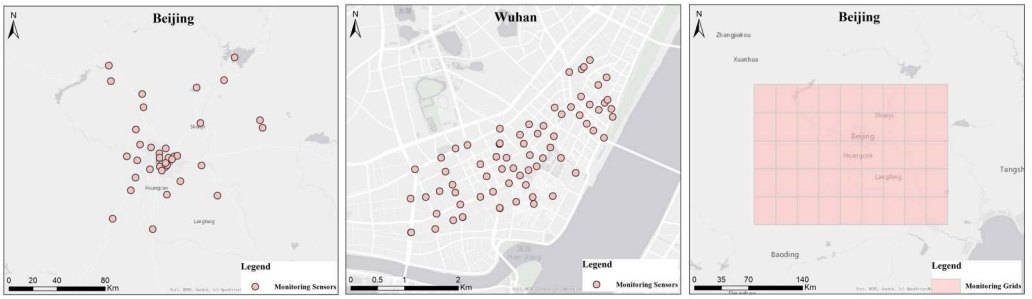

Figure 5. Spatial distribution of three datasets: (a) $PM_{2.5}$ dataset, (b)traffic dataset, and (b) temperature dataset.

(3) We partitioned the spatio-temporal data into training and testing samples, with the training samples constituting 80% and the testing samples 20%.

5.2. Evaluation metrics and baselines

5.2.1. Performance criteria

This study mainly evaluate the three indicators of the model, i.e. prediction precision, computational efficiency, and scale of learnable parameters. More specifically, we first utilized RMSE and MAPE to quantitatively measure the prediction precision of the PSDCN model. Then, we evaluated the computational efficiency by calculating the runtime for both forward and backward propagation. Finally, we determined the scale of learnable parameters by counting the number of trainable parameters. Considering that the computation of runtime and parameter count is straightforward, we only present the calculation methods for prediction precision metrics, as shown in Equations (11)—(12).

$$RMSE = \sqrt{\frac{1}{n*q}\sum_{i=1}^{n}\sum_{j=1}^{q}\left(x_i^{t+j} - \hat{x}_i^{t+j}\right)^2} \tag{11}$$

$$MAPE = \frac{100}{n*q}\sum_{i=1}^{n}\sum_{j=1}^{q}\left|\frac{x_i^{t+j} - \hat{x}_i^{t+j}}{x_i^{t+j}}\right| \tag{12}$$

where x_i^{t+j} and $\hat{x}_i^{t+j}$ respectively indicate the actual values and predicted value of node v_i in the $(t+j)$-th time window; n denotes the total count of graph nodes; and q denotes the prediction horizon.

5.2.2. Baselines

In this study, nine baseline models were employed to analyze the strengths of the PSDCN models, organized into three categories.

- **Statistical learning or machine learning models**: The first category emphasizes the advantages of deep learning models in terms of prediction precision, primarily including HA model(Campbell and Thompson 2008), the ST-KNN model(Z. Zheng and Su 2014), and the BTMF model (X. Chen and Sun 2022).
- **Deep learning models without lightweight design:** The second category emphasizes the advantages of the proposed PSDCN model in terms of both computational efficiency and prediction precision, primarily including the ST-GCN model(Yu, Yin, and Zhu 2018), the BiSTGN model(Wang et al. 2022), the DSTAGNN model (Lan et al. 2022), the STA-ODE model (P. Wang, Zhang, Zhang, et al. 2024), and the GDGCN model(Y. Xu et al. 2023).

- **Deep learning models with lightweight design:** The third category emphasizes the advantages of the proposed PSDCN model in terms of parameter scale, primarily including the STGDN model (P. Wang, Zhang, Cheng, et al. 2024).

5.3. Hyper-parameter calibration

The hyper-parameters of the PSDCN model mainly include the historical dependency horizon p, the dimension f of the hidden layer, the number L of spatio-temporal blocks, and the kernel size K of the PSTDC operator, the dilation factor d_l^T/d_l^S of the PSTDC/PSGDC operator in the l-th spatio-temporal block. Theoretically, each hyper-parameter of the PSDCN model must be calibrated using the control variable method. However, by utilizing the hyper-parameter dependency component, we can calibrate multiple hyper-parameters simultaneously. For example, Equation (7) allows for the simultaneous calibration of the dilation factor d_l^T and the kernel size K, while Equation (9) facilitates the simultaneous calibration of the historical dependency horizon p and the number L of spatio-temporal blocks.

Based on the previous analysis, the PSDCN model requires calibration of only three hyperparameters: the historical dependency horizon p, the dimension f of the hidden layer, and the kernel size K of the PSTDC operator. To further simplify hyper-parameter tuning, we fixed the historical dependency horizon at 10 for all three datasets. Taking the $PM_{2.5}$ dataset as an example, Figure 6 shows the calibration process of the PSDCN model. Figure 6(a) indicates that the RMSE initially decreases and then stabilizes as f increases. In general, an increase in f enhances the model's fitting ability while also increasing its parameter scale. Therefore, setting f to 32 is a highly appropriate choice. Figure 6(b) indicates that the RMSE first stabilizes and then increases as K increases. From Equation (7), it can be observed that the dilation factor increases as K increases. While this enhancement in the dilation factor improves the computational efficiency of the model, it also results in the loss of certain information, accounting for the decrease in model accuracy as K increases. Therefore, setting K to 2 is a highly appropriate choice. Similarly, for the traffic and temperature datasets, f and K were determined in the same manner as described above, with f ultimately set to 32 and K set to 2.

5.4. Quantitative analysis

This subsection first compared the differences in prediction precision between the PSDCN model and baselines. Then, we compared the differences in computational efficiency between the PSDCN

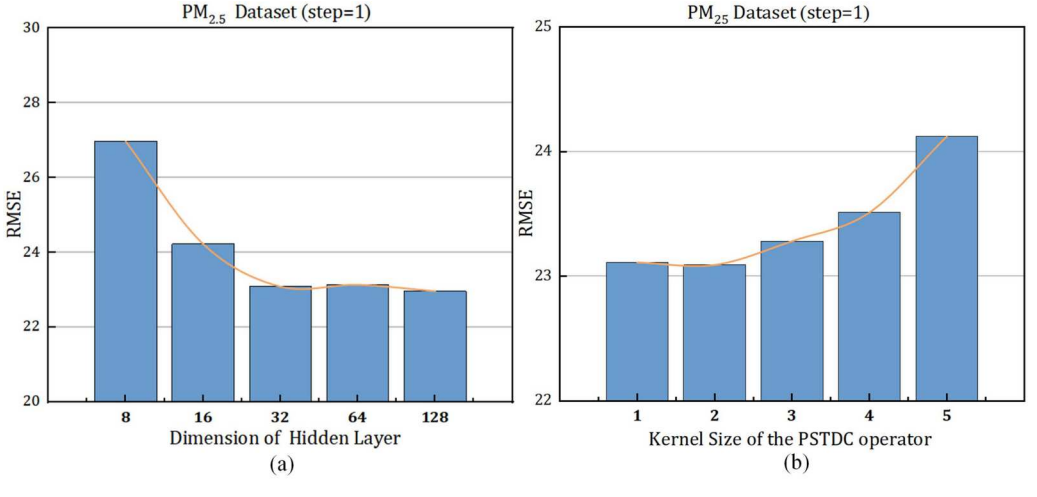

Figure 6. Hyper-parameter tuning of the PSDCN model on the $PM_{2.5}$ dataset: (a) the dimension of the hidden layer, and (b) the kernel size of the PSTDC operator.

model and baselines. Finally, we compared the differences in the scale of learnable parameters between the PSDCN model and baselines.

5.4.1. Quantitative analysis of prediction precision

During model training, different random seeds initialize distinct learnable parameters, which leads to subtle variations in prediction precision of the model. To fairly compare the prediction precision of the PSDCN model and baselines, we present the optimal metrics of the models under different random seeds, as summarized in Table 1. The results indicate that the prediction precisions of the second and third category models exceed those of the first category models. Specifically, the deep learning models outperform both the machine learning and statistical learning models in terms of prediction precision. The reason is that deep learning models exhibit superior nonlinear fitting capabilities, making them more suited for spatio-temporal prediction tasks. Additionally, the results show that the prediction precision of the proposed PSDCN model is comparable to that of existing advanced deep learning models, such as the DSTAGNN model, the STA-ODE model, the GDGCN model, and the ST-GDN model. The reason is that the proposed PSDCN improves computational efficiency and reduces the parameter scale without sacrificing prediction precision. Furthermore, the results indicate that the prediction accuracy of the PSDCN model and baselines varies across the three datasets. This variation primarily stems from differences in the complexity of the spatio-temporal dependencies within each dataset. For example, temperature data generally exhibit smoother changes, while traffic and $PM_{2.5}$ data may experience abrupt fluctuations. Based on the t-test and the 5-step prediction results, Table 2 illustrates the statistical significance of the predictions. The results indicate that the proposed PSDCN significantly outperformed the baselines in most cases. Only one group out of 12 tests failed the hypothesis test at the 5% significance level. Figure 7 illustrates the stability of prediction models with superior precision. The results show that the PSDCN model maintains relatively stable prediction precision across different random seeds, further affirming its ability to compete with advanced prediction models.

Table 1. Prediction precision (RMSE/MAPE) of the PSDCN model and baselines.

	$PM_{2.5}$ Dataset		Traffic Dataset		Temperature Dataset	
Models	1-step	5-steps	1-step	5-steps	1-step	5-steps
HA	46.36/72.32	73.80/141.1	8.17/20.32	41.10/281.79	3.81/11.60	5.56/16.97
ST-KNN	41.11/62.66	54.31/102.5	8.69/24.25	10.17/27.08	1.94/5.52	2.84/7.01
BTMF	32.23/50.04	51.81/95.3	10.08/35.4	9.63/25.94	1.49/4.29	2.44/6.88
ST-GCN	28.46/39.47	50.17/89.54	8.67/24.73	9.78/28.67	1.38/3.72	2.34/6.42
BiSTGN	24.99/31.56	49.38/83.06	8.32/20.41	9.02/25.69	1.17/3.18	2.38/6.09
DSTAGNN	23.32/28.70	48.12/80.67	7.05/19.32	8.71/23.27	1.31/3.37	2.24/6.24
STA-ODE	23.78/28.99	47.95/80.37	7.09/19.97	8.82/24.73	1.07/2.88	2.19/5.83
GDGCN	23.43/29.70	47.23/79.43	7.03/19.61	8.69/23.82	1.04/2.72	2.12/5.74
STGDN	23.10/28.52	47.17/78.78	7.03/19.09	8.65/22.78	1.03/2.72	2.09/5.73
PSDCN	23.09/28.43	47.16/78.71	7.05/18.72	8.63/22.52	1.04/2.72	2.06/5.71

Table 2. Statistical significance based on *t*-test.

	$PM_{2.5}$ Dataset		Traffic Dataset		Temperature Dataset	
Models	*t*-statistic	*p*-value	*t*-statistic	*p*-value	*t*-statistic	*p*-value
DSTAGNN	−6.84	0.001	−3.82	0.005	−4.55	0.002
STA-ODE	−4.09	0.003	2.03	0.076	−5.26	0.001
GDGCN	−2.91	0.012	2.66	0.028	−3.01	0.016
STGDN	−1.79	0.097	2.82	0.022	−2.39	0.043

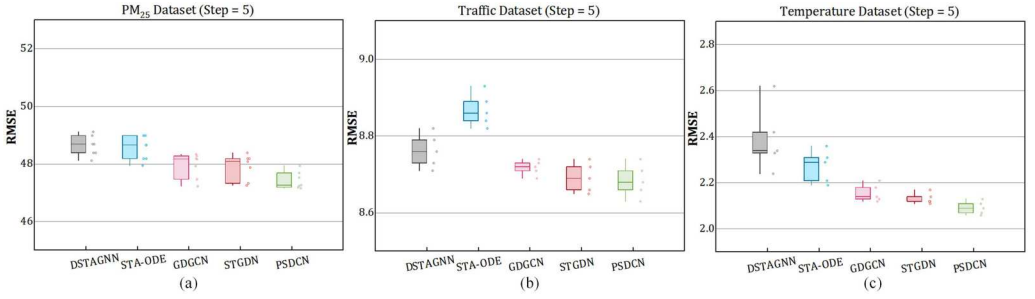

Figure 7. Stability of the PSDCN model and baselines across different random seeds: (a) $PM_{2.5}$ dataset, (b) traffic dataset, and (c) temperature dataset.

5.4.2. Quantitative analysis of computational efficiency

In addition to evaluating prediction precision, we conducted a comprehensive analysis of the computational efficiency between the PSDCN model and various baselines. Given that the proposed PSDCN model is a deep learning architecture, our comparisons were focused on deep learning models, as summarized in Table 3. In this context, inference time reflects the speed of online predictions, while optimization time indicates the efficiency of offline training. The results suggest the PSDCN model exhibits superior computational efficiency compared to baselines such as ST-GCN, DSTAGNN, STA-ODE, GDGCN, and STGDN. This can be attributed to the fact that the PSDCN model carefully adjusts the number of forward propagation using the dilation operation and hyper-parameter dependency component, reducing the running time of model. The results also indicate that the computational efficiency of the PSDCN model and the baselines varies slightly across the three datasets. This variation is attributed to differences in the number of graph nodes in each dataset. Specifically, as the number of graph nodes decreases from the traffic dataset to the temperature dataset and then to the $PM_{2.5}$ dataset, the computational efficiency increases accordingly. Additionally, Figure 8 shows that the PSDCN model exhibits a slower running time than the BiSTGN model, yet achieves higher prediction precision. Notably, the computational efficiency of the proposed PSDCN model significantly exceeds that of existing lightweight STGDN model. The reason may be that the overly

Table 3. Running time (milliseconds) of the PSDCN model and baselines with batch size being 256.

Models		Time for One Inference	Time for One Optimization
ST-GCN	$PM_{2.5}$ Dataset	40.29±2.94	55.18±4.18
	Traffic Dataset	276.47 ± 19.18	368.31 ± 20.18
	Temperature Dataset	141.14 ± 15.47	171.24 ± 16.87
BiSTGN	$PM_{2.5}$ Dataset	29.84 ± 2.29	45.20 ± 2.29
	Traffic Dataset	49.87 ± 3.88	66.38 ± 4.73
	Temperature Dataset	34.19 ± 2.22	49.40 ± 2.80
DSTAGNN	$PM_{2.5}$ Dataset	38.62 ± 3.74	49.29 ± 4.55
	Traffic Dataset	224.39 ± 7.79	325.84 ± 9.29
	Temperature Dataset	123.74 ± 14.73	158.51 ± 15.16
STA-ODE	$PM_{2.5}$ Dataset	135.21 ± 11.78	180.17 ± 14.24
	Traffic Dataset	987.14 ± 30.78	1253.1 ± 35.15
	Temperature Dataset	367.98 ± 19.76	434.42 ± 21.67
GDGCN	$PM_{2.5}$ Dataset	71.84 ± 2.64	171.10 ± 14.24
	Traffic Dataset	208.66 ± 10.12	575.55 ± 25.35
	Temperature Dataset	84.77 ± 3.94	304.19 ± 19.17
STGDN	$PM_{2.5}$ Dataset	27.26 ± 1.23	48.10 ± 1.38
	Traffic Dataset	173.73 ± 15.38	258.77 ± 17.28
	Temperature Dataset	59.47 ± 5.76	93.03 ± 6.16
PSDCN	$PM_{2.5}$ Dataset	20.91 ± 1.75	35.90 ± 2.36
	Traffic Dataset	32.46 ± 3.84	55.83 ± 4.41
	Temperature Dataset	17.95 ± 1.85	31.92 ± 5.57

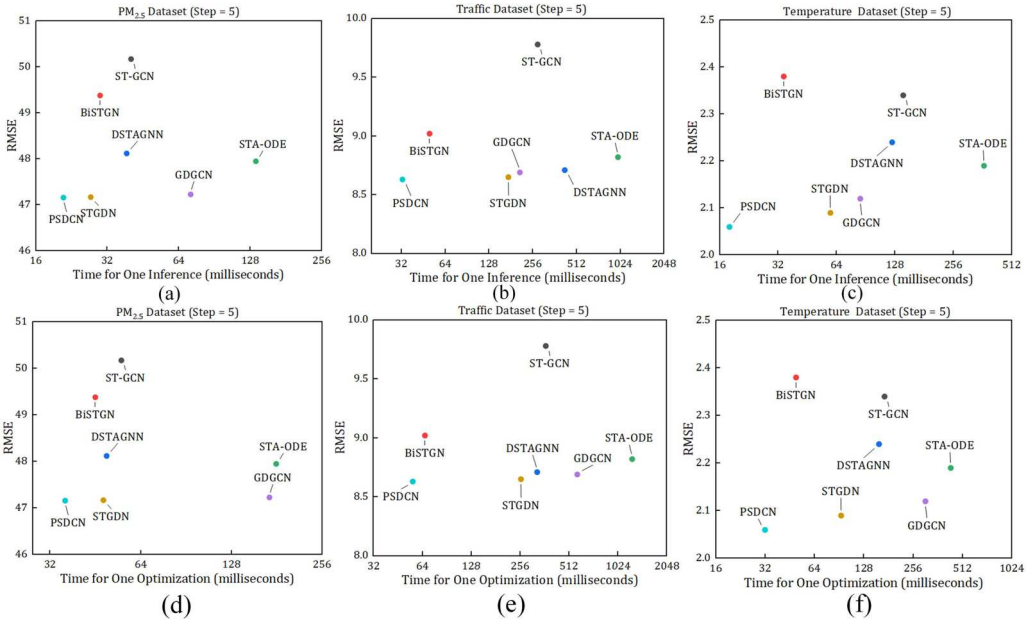

Figure 8. Prediction precision vs. computational efficiency: (a) inference time on $PM_{2.5}$ dataset, (b) inference time on traffic dataset, (c) inference time on temperature dataset, (d) optimization time on $PM_{2.5}$ dataset, (e) optimization time on traffic dataset, and (f) optimization time on temperature dataset.

complex parameter adjustment process in the STGDN model makes it challenging to optimize computational efficiency while maintaining prediction accuracy. However, the hyper-parameter dependency component introduced in this study effectively addresses this challenge.

5.4.3. Quantitative analysis of parameter scale

Similar to computational efficiency, we examined the scale of learnable parameters, as detailed in Table 4. Our investigation highlights a significant advantage of the PSDCN model over baseline models regarding the scale of learnable parameters. Moreover, when compared with the lightweight STGDN model, the PSDCN model also achieves a reduction in the scale of its learnable parameters. This advantage arises from the substantial reduction in parameter scale achieved through the parameter-sharing mechanism. The results also demonstrate that the PSDCN model maintains a consistent parameter scale across all three datasets. The primary reason is that the parameter sizes of the PSDCN model are primarily determined by the dimensions of the hidden layers. Since the hidden layer sizes are identical across the three datasets, this results in an equal number of learnable parameters in the model. Furthermore, Figure 9 presents a scatter plot that visualizes the relationship between parameter scale and prediction precision. Despite having fewer learnable parameters, the PSDCN model maintains superior prediction precision.

Table 4. Parameter Scale (bytes) of learnable weights for the PSDCN models and baselines.

Models	$PM_{2.5}$ Dataset	Traffic Dataset	Temperature Dataset
ST-GCN	96209	171773	96245
BiSTGN	13509	29477	13249
DSTAGNN	70457	113749	92221
STA-ODE	286138	605572	423653
GDGCN	198980	277580	273420
STGDN	38149	35077	22469
PSDCN	8613	8613	8613

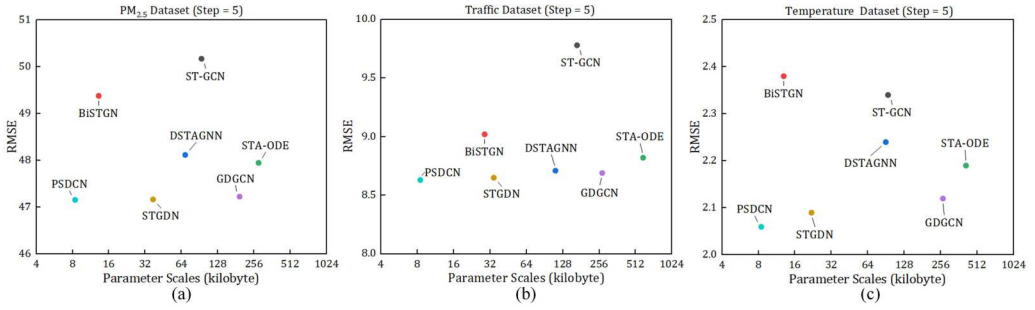

Figure 9. Prediction precision vs. scale of learnable parameters: (a) $PM_{2.5}$ dataset, (b) traffic dataset, and (c) temperature dataset.

5.5. Qualitative analysis

This subsection visually illustrates the prediction precision of the PSDCN model. Figure 10 displays discrepancies between predicted values and actual values in the temporal dimension, while Figure 11 displays variations in the spatial dimension. Our findings reveal that the PSDCN model shows notably lower prediction precision for multi-step forecasting compared to single-step forecasting, particularly in time intervals and spatial areas with significant data fluctuations. This observation aligns with existing research indicating that irregular fluctuations in spatio-temporal data can affect prediction precision (Wang, Zhang, Cheng, et al., 2024). Therefore, the observed decrease in prediction precision of the PSDCN model is consistent with these expectations. Furthermore, despite the marked variation in prediction precision between single-step and multi-step forecasting, the proposed PSDCN model consistently captures temporal trends and relative spatial distributions of data accurately, underscoring its strong prediction performance. Leveraging the proven advantages of PSDCN in computational efficiency and parameter scalability, we can deploy the model on resource-constrained devices to support managerial decision-making. For example, rapid traffic

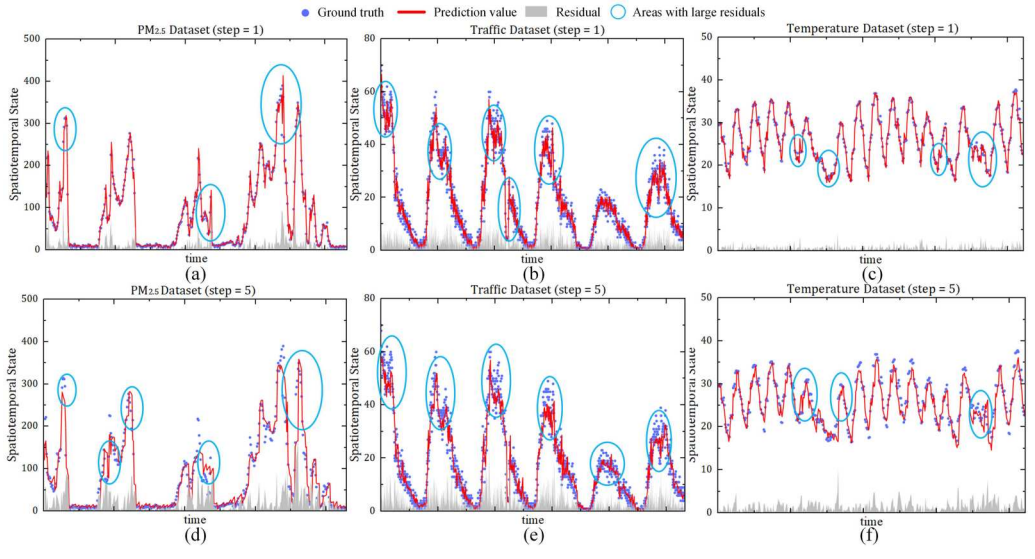

Figure 10. Temporal visualization depicting the predicted results of the PSDCN model: (a) predicting one-step on $PM_{2.5}$ dataset, (b) predicting five-steps on $PM_{2.5}$ dataset, (c) predicting one-step on traffic dataset, (d) predicting five-steps on traffic dataset, (e) predicting one-step on temperature dataset, and (f) predicting five-steps on temperature dataset.

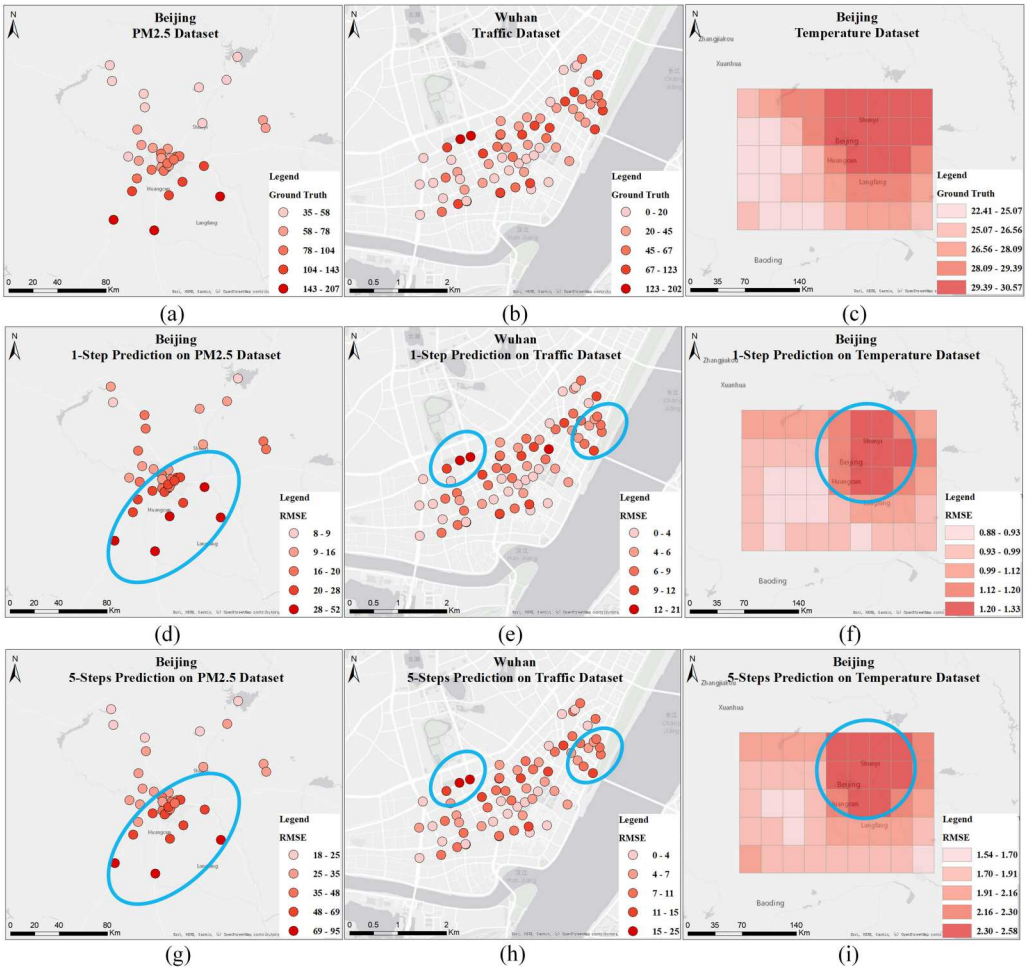

Figure 11. Spatial visualization depicting the predicted results of the PSDCN model: (a) actual values on $PM_{2.5}$ dataset, (b) actual values on $PM_{2.5}$ dataset, (c) actual values on traffic dataset, (d) predicting one-step on $PM_{2.5}$ dataset, (e) predicting five-steps on $PM_{2.5}$ dataset, (f) predicting one-step on traffic dataset, (g) predicting five-steps on traffic dataset, (h) predicting one-step on temperature dataset, and (i) predicting five-steps on temperature dataset.

flow prediction can enable precise traffic diversion, while quick air quality prediction can issue timely alerts to safeguard urban residents from pollution.

5.6. Ablation study

In this subsection, ablation studies are conducted to validate the rationale behind the model design. Specifically, Table 5 presents the prediction precision of the PSDCN model and its components,

Table 5. Prediction precision (RMSE/MAPE) between the PSDCN model and its components.

	$PM_{2.5}$ Dataset		Traffic Dataset		Temperature Dataset	
Models	1-step	5-steps	1-step	5-steps	1-step	5-steps
PSTDC	24.15/34.32	48.13/82.71	7.53/22.09	8.71/24.32	1.19/3.04	2.19/5.91
PSGDC	25.12/35.12	49.13/84.12	8.23/24.02	8.84/25.12	1.24/3.27	2.31/6.08
PSDCN	23.09/28.43	47.16/78.71	7.05/18.72	8.63/22.52	1.04/2.72	2.06/5.71

Table 6. Impact of parameter sharing and dilated operations on model predictions.

Designs		Precision ($PM_{2.5}$:RMSE/MAPE)	Efficiency (inference/optimization)	Scale (bytes)
Dilatated Operations	Y	23.09/28.43	20.91/35.90	—
	N	23.03/28.07	32.43/51.37	—
Parameter Sharing	Y	23.09/28.43	—	8613
	N	22.93/27.69	—	18913

while Table 6 illustrates the impact of parameter sharing and dilated operations on model performance. The results indicate that the prediction precision of the PSTDC component exceeds that of the PSGDC component, and the PSDCN model outperforms both individual components—PSTDC and PSGDC—in terms of prediction precision. These findings highlight the significant impact of temporal correlation on prediction outcomes, underscoring the necessity of integrating both temporal and spatial relationships simultaneously to improve the overall prediction performance of the model. In addition, the results indicate that the dilatated operation enhances the computational efficiency of the model without sacrificing prediction precision. On the other hand, parameter sharing significantly reduces the model's parameter scale, although it may lead to a slight loss in prediction precision. These findings support the use of dilatated operation and parameter sharing.

6. Discussions and conclusions

Lightweight and high-precision prediction models for urban states are anticipated to run efficiently on resource-limited devices, serving as key technologies for realizing smart city management. However, many existing models, despite achieving high prediction precision, suffer from overly complex designs, leading to low computational efficiency, a large number of learnable parameters, and difficulty in hyper-parameter calibration. To address these challenges, we establish a lightweight spatio-temporal prediction model, namely the PSDCN model.

In the experimental section, the proposed PSDCN model was validated using $PM_{2.5}$, traffic, and temperature datasets. Compared to deep learning models with superior precision (Lan et al. 2022; P. Wang, Zhang, Zhang, et al. 2024; Y. Xu et al. 2023), the proposed PSDCN model not only demonstrates efficient computational performance but also achieves or approaches the prediction precision of various baselines, indicating that the PSDCN model can be applied to scenarios requiring fast model training and inference speed. In comparison to existing lightweight models (P. Wang, Zhang, Cheng, et al. 2024), the PSDCN model strictly controls the scale of learnable parameters and the number of hyper-parameters, reducing both the spatial complexity and the training difficulty of the model. Additionally, we conducted ablation studies to validate the rationale behind the parameter-sharing mechanism and dilated operation. Overall, we believe that the PSDCN model is a valuable spatio-temporal prediction model.

Given that many existing prediction models are constrained by complex designs, the proposed parameter-sharing mechanism and dilated operation offer substantial potential for widespread application. On one hand, we can integrate the concepts of parameter sharing and dilated operation into complex models to simplify the existing network structure without compromising prediction precision (Guangyue Li et al. 2024; Y. Xu et al. 2023). On the other hand, the PSDCN model can serve as a base learner for ensemble learning, enabling the construction of a lightweight ensemble learning model to further enhance prediction precision (Cheng et al. 2019; Jia and Yan 2021).

There are two limitations in this study: (1) Similar to existing graph-based prediction models, the proposed PSDCN model faces computational bottlenecks when dealing with large graph structures, especially those with over ten thousand nodes; (2) The proposed PSDCN can be regarded as a base operator, resulting in suboptimal performance in highly complex scenarios, such as those involving missing data. To address the aforementioned limitations, future work will concentrate on two key aspects. First, we will examine the critical value for the number of graph nodes and identify the maximum number of nodes to which the PSDCN model can be applied. Second, we will enhance

the existing missing-data-tolerant model by integrating the ideas presented in this work, with the goal of developing a lightweight, high-precision prediction model for complex scenarios.

Acknowledgments

The numerical calculations in this paper have been done on the supercomputing system in the Supercomputing Center of Wuhan University.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This project was supported by Supported by the National Key Research and Development Program of China [grant number 2021YFB3900803]; National Natural Science Foundation of China [Grant Nos. 42401524, 42371469, and 42371470]; China National Postdoctoral Support Program for Innovative Talents [grant number BX20230360]; China Postdoctoral Science Foundation [grant number 2023M743454]; Open Fund of National Engineering Research Center of Geographic Information System, China University of Geosciences, Wuhan 430074, China [grant number 2023KFJJ09].

Data availability statement

The data and codes supporting the main findings of this study are available at anonymous private link on https://doi.org/10.6084/m9.figshare.26028223.

Geolocation statement

Internal GPS chips were used to determine location.

ORCID

Peixiao Wang http://orcid.org/0000-0002-1209-6340
Haolong Yang http://orcid.org/0009-0001-9775-8285
Hengcai Zhang http://orcid.org/0000-0002-5004-9609
Shifen Cheng http://orcid.org/0000-0002-9553-8318
Feng Lu http://orcid.org/0000-0001-6573-2550
Zeqiang Chen http://orcid.org/0000-0001-6624-6693

References

Campbell, John Y., and Samuel B. Thompson. 2008. "Predicting Excess Stock Returns Out of Sample: Can Anything Beat the Historical Average?" *Review of Financial Studies* 21 (4): 1509–1531. https://doi.org/10.1093/rfs/hhm055.

Chen, Ricky T. Q., Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. 2018. "Neural Ordinary Differential Equations." In Proceedings of the 32nd International Conference on Neural Information Processing Systems, 6572–6583. arXiv. https://doi.org/10.48550/arXiv.1806.07366.

Chen, X., and L. Sun. 2022. "Bayesian Temporal Factorization for Multidimensional Time Series Prediction." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44:4659–4673. https://doi.org/10.1109/TPAMI.2021.3066551.

Cheng, Shifen, Feng Lu, Peng Peng, and Sheng Wu. 2019. "Multi-Task and Multi-View Learning Based on Particle Swarm Optimization for Short-Term Traffic Forecasting." *Knowledge-Based Systems* 180:116–132. https://doi.org/10.1016/j.knosys.2019.05.023.

Cheng, Shifen, Peng Peng, and Feng Lu. 2020. "A Lightweight Ensemble Spatiotemporal Interpolation Model for Geospatial Data." *International Journal of Geographical Information Science* 34 (9): 1849–1872. https://doi.org/10.1080/13658816.2020.1725016.

Cheng, Shifen, Lizeng Wang, Peixiao Wang, and Feng Lu. 2024. "An Ensemble Spatial Prediction Method Considering Geospatial Heterogeneity." *International Journal of Geographical Information Science* 38 (9): 1856–1880. https://doi.org/10.1080/13658816.2024.2358052.

Chien, Wei-Che, and Yueh-Min Huang. 2021. "A Lightweight Model with Spatial–Temporal Correlation for Cellular Traffic Prediction in Internet of Things." *The Journal of Supercomputing* 77 (9): 10023–10039. https://doi.org/10.1007/s11227-021-03662-2.

Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." arXiv:1412.3555 [Cs], December. http://arxiv.org/abs/1412.3555.

Do, Loan N. N., Hai L. Vu, Bao Q. Vo, Zhiyuan Liu, and Dinh Phung. 2019. "An Effective Spatial-Temporal Attention Based Neural Network for Traffic Flow Prediction." *Transportation Research Part C: Emerging Technologies* 108:12–28. https://doi.org/10.1016/j.trc.2019.09.008.

Fang, Zheng, Qingqing Long, Guojie Song, and Kunqing Xie. 2021. "Spatial-Temporal Graph ODE Networks for Traffic Flow Forecasting." In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 364–373. KDD '21. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3447548.3467430.

Guan, Qingfeng, Jingyi Wang, Shuliang Ren, Huan Gao, Zhewei Liang, Junyi Wang, and Yao Yao. 2024. "Predicting Short-Term PM2.5 Concentrations at Fine Temporal Resolutions Using a Multi-Branch Temporal Graph Convolutional Neural Network." *International Journal of Geographical Information Science* 38 (4): 778–801. https://doi.org/10.1080/13658816.2024.2310737.

Guo, Shengnan, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. "Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting." *Proceedings of the AAAI Conference on Artificial Intelligence* 33:922–929. https://doi.org/10.1609/aaai.v33i01.3301922.

Jia, Tao, and Penggao Yan. 2021. "Predicting Citywide Road Traffic Flow Using Deep Spatiotemporal Neural Networks." *IEEE Transactions on Intelligent Transportation Systems* 22 (5): 3101–3111. https://doi.org/10.1109/TITS.2020.2979634.

Karl, Matthias, Sina Acksen, Rehan Chaudhary, and Martin O. P. Ramacher. 2024. "Forecasting System for Urban Air Quality with Automatic Correction and Web Service for Public Dissemination." *International Journal of Digital Earth* 17 (1): 1–22. https://doi.org/10.1080/17538947.2024.2359569.

Kipf, Thomas N., and Max Welling. 2017. "Semi-Supervised Classification with Graph Convolutional Networks." In *International Conference on Learning Representations*. http://arxiv.org/abs/1609.02907.

Lan, Shiyong, Yitong Ma, Weikang Huang, Wenwu Wang, Hongyu Yang, and Pyang Li. 2022. "DSTAGNN: Dynamic Spatial-Temporal Aware Graph Neural Network for Traffic Flow Forecasting." In Proceedings of the 39th International Conference on Machine Learning, 11906–11917. PMLR. https://proceedings.mlr.press/v162/lan22a.html.

Li, Qiang, Qi Wang, and Xuelong Li. 2021. "Exploring the Relationship Between 2D/3D Convolution for Hyperspectral Image Super-Resolution." *IEEE Transactions on Geoscience and Remote Sensing* 59 (10): 8693–8703. https://doi.org/10.1109/TGRS.2020.3047363.

Li, Guangyue, Zilong Zhao, Xiaogang Guo, Luliang Tang, Huazu Zhang, and Jinghan Wang. 2024. "Towards Integrated and Fine-Grained Traffic Forecasting: A Spatio-Temporal Heterogeneous Graph Transformer Approach." *Information Fusion* 102:102063. https://doi.org/10.1016/j.inffus.2023.102063.

Li, Guanyao, Shuhan Zhong, Xingdong Deng, Letian Xiang, S.-H. Gary Chan, Ruiyuan Li, Yang Liu, Ming Zhang, Chih-Chieh Hung, and Wen-Chih Peng. 2023. "A Lightweight and Accurate Spatial-Temporal Transformer for Traffic Forecasting." *IEEE Transactions on Knowledge and Data Engineering* 35 (11): 10967–10980. https://doi.org/10.1109/TKDE.2022.3233086.

Liang, Yuxuan, Yutong Xia, Songyu Ke, Yiwei Wang, Qingsong Wen, Junbo Zhang, Yu Zheng, and Roger Zimmermann. 2023. "AirFormer: Predicting Nationwide Air Quality in China with Transformers." *Proceedings of the AAAI Conference on Artificial Intelligence* 37 (12): 14329–14337. https://doi.org/10.1609/aaai.v37i12.26676.

Liu, Shang, Miao He, Zhiqiang Wu, Peng Lu, and Weixi Gu. 2023. "Spatial–Temporal Graph Neural Network Traffic Prediction Based Load Balancing with Reinforcement Learning in Cellular Networks." *Information Fusion* 103:102079. https://doi.org/10.1016/j.inffus.2023.102079.

Liu, Yutian, Soora Rasouli, Melvin Wong, Tao Feng, and Tianjin Huang. 2024. "RT-GCN: Gaussian-Based Spatiotemporal Graph Convolutional Network for Robust Traffic Prediction." *Information Fusion* 102:102078. https://doi.org/10.1016/j.inffus.2023.102078.

Mengfan, Teng, Li Siwei, Song Ge, Yang Jie, Dong Lechao, Lin Hao, and Hu Senlin. 2022. "Including the Feature of Appropriate Adjacent Sites Improves the PM2.5 Concentration Prediction with Long Short-Term Memory Neural Network Model." *Sustainable Cities and Society* 76:103427. https://doi.org/10.1016/j.scs.2021.103427.

Shi, Xingjian, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. 2017. "Deep Learning for Precipitation Nowcasting: A Benchmark and A New Model," 18.

Tan, Cheng, Zhangyang Gao, Lirong Wu, Yongjie Xu, Jun Xia, Siyuan Li, and Stan Z. Li. 2023. "Temporal Attention Unit: Towards Efficient Spatiotemporal Predictive Learning." In, 18770–18782. https://openaccess.thecvf.com/

content/CVPR2023/html/Tan_Temporal_Attention_Unit_Towards_Efficient_Spatiotemporal_Predictive_Learning_CVPR_2023_paper.html.

Wang, Jianlong, Xiaoqi Duan, Peixiao Wang, A-Gen Qiu, and Zeqiang Chen. 2024. "Predicting Urban Signal-Controlled Intersection Congestion Events Using Spatio-Temporal Neural Point Process." *International Journal of Digital Earth* 17 (1): 2376270. https://doi.org/10.1080/17538947.2024.2376270.

Wang, Huiliang, Shanlun Xu, Hongshi Xu, Zening Wu, Tianye Wang, and Chao Ma. 2023. "Rapid Prediction of Urban Flood Based on Disaster-Breeding Environment Clustering and Bayesian Optimized Deep Learning Model in the Coastal City." *Sustainable Cities and Society* 99:104898. https://doi.org/10.1016/j.scs.2023.104898.

Wang, Peixiao, Hengcai Zhang, Shifen Cheng, Tong Zhang, Feng Lu, and Sheng Wu. 2024. "A Lightweight Spatiotemporal Graph Dilated Convolutional Network for Urban Sensor State Prediction." *Sustainable Cities and Society* 101:105105. https://doi.org/10.1016/j.scs.2023.105105.

Wang, Peixiao, Yan Zhang, Tao Hu, and Tong Zhang. 2023. "Urban Traffic Flow Prediction: A Dynamic Temporal Graph Network Considering Missing Values." *International Journal of Geographical Information Science* 37 (4): 885–912. https://doi.org/10.1080/13658816.2022.2146120.

Wang, Peixiao, Hengcai Zhang, Jie Liu, Feng Lu, and Tong Zhang. 2025. "Efficient Inference of Large-Scale Air Quality Using a Lightweight Ensemble Predictor." *International Journal of Geographical Information Science* 39: 1–25. https://doi.org/10.1080/13658816.2024.2437044.

Wang, Peixiao, Tong Zhang, Hengcai Zhang, Shifen Cheng, and Wangshu Wang. 2024. "Adding Attention to the Neural Ordinary Differential Equation for Spatio-Temporal Prediction." *International Journal of Geographical Information Science* 38 (1): 156–181. https://doi.org/10.1080/13658816.2023.2275160.

Wang, Peixiao, Tong Zhang, Yueming Zheng, and Tao Hu. 2022. "A Multi-View Bidirectional Spatiotemporal Graph Network for Urban Traffic Flow Imputation." *International Journal of Geographical Information Science* 36 (6): 1231–1257. https://doi.org/10.1080/13658816.2022.2032081.

Xie, Peng, Tianrui Li, Jia Liu, Shengdong Du, Xin Yang, and Junbo Zhang. 2020. "Urban Flow Prediction from Spatiotemporal Data Using Machine Learning: A Survey." *Information Fusion* 59:1–12. https://doi.org/10.1016/j.inffus.2020.01.002.

Xu, Mingxing, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. 2021. "Spatial-Temporal Transformer Networks for Traffic Flow Forecasting." arXiv:2001.02908 [Cs, Eess], March. http://arxiv.org/abs/2001.02908.

Xu, Yi, Liangzhe Han, Tongyu Zhu, Leilei Sun, Bowen Du, and Weifeng Lv. 2023. "Generic Dynamic Graph Convolutional Network for Traffic Flow Forecasting." *Information Fusion* 100: 101946. https://doi.org/10.1016/j.inffus.2023.101946.

Yang, Li, and Abdallah Shami. 2020. "On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice." *Neurocomputing* 415: 295–316. https://doi.org/10.1016/j.neucom.2020.07.061.

Yu, Bing, Haoteng Yin, and Zhanxing Zhu. 2018. "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting." In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 3634–3640. Stockholm, Sweden: International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2018/505.

Zhang, Tong, Jie Liu, Chulin Gao, Peixiao Wang, Liang Leng, and Yanjiao Xiao. 2024. "Prior-Guided Gated Convolutional Networks for Rainstorm Forecasting." *Journal of Hydrology* 633:130962. doi:https://doi.org/10.1016j.jhydrol.2024.130962.

Zhang, Tong, Jie Liu, and Jianlong Wang. 2022. "Rainstorm Prediction via a Deep Spatio-Temporal-Attributed Affinity Network." *Geocarto International* 37 (26). 13079–13097. https://doi.org/10.1080/10106049.2022.2076914.

Zhang, Yuchen, Mingsheng Long, Kaiyuan Chen, Lanxiang Xing, Ronghua Jin, Michael I. Jordan, and Jianmin Wang. 2023. "Skilful Nowcasting of Extreme Precipitation with NowcastNet." *Nature* 619 (7970): 526–532. https://doi.org/10.1038/s41586-023-06184-4.

Zhang, Lei, Jiaming Na, Jie Zhu, Zhikuan Shi, Changxin Zou, and Lin Yang. 2021. "Spatiotemporal Causal Convolutional Network for Forecasting Hourly PM2.5 Concentrations in Beijing, China." *Computers & Geosciences* 155: 104869. https://doi.org/10.1016/j.cageo.2021.104869.

Zhang, Tong, Jianlong Wang, Tong Wang, Yiwei Pang, Peixiao Wang, and Wangshu Wang. 2024. "A Deep Marked Graph Process Model for Citywide Traffic Congestion Forecasting." *Computer-Aided Civil and Infrastructure Engineering* 39 (8): 1180–1196. https://doi.org/10.1111/mice.13131.

Zhang, Weishi, Ying Xu, David G. Streets, and Can Wang. 2025. "Measurement of Urbanization and Its Spatiotemporal Heterogenous Effects on Carbon Emission from District Heating Industry in China." *Energy and Buildings* 328:115182. https://doi.org/10.1016/j.enbuild.2024.115182.

Zhao, Ling, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2020. "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction." *IEEE Transactions on Intelligent Transportation Systems* 21 (9): 3848–3858. https://doi.org/10.1109/TITS.2019.2935152.

Zheng, Ge, Wei Koong Chai, Jing-Lin Duanmu, and Vasilis Katos. 2023. "Hybrid Deep Learning Models for Traffic Prediction in Large-Scale Road Networks." *Information Fusion* 92:93–114. https://doi.org/10.1016/j.inffus.2022.11.019.

Zheng, Zuduo, and Dongcai Su. 2014. "Short-Term Traffic Volume Forecasting: A k-Nearest Neighbor Approach Enhanced by Constrained Linearly Sewing Principle Component Algorithm." *Transportation Research Part C: Emerging Technologies*, Special Issue on Short-term Traffic Flow Forecasting 43:143–157. https://doi.org/10.1016/j.trc.2014.02.009.

Zheng, Qinghe, Xinyu Tian, Zhiguo Yu, Nan Jiang, Abdussalam Elhanashi, Sergio Saponara, and Rui Yu. 2023. "Application of Wavelet-Packet Transform Driven Deep Learning Method in PM2.5 Concentration Prediction: A Case Study of Qingdao, China." *Sustainable Cities and Society* 92:104486. https://doi.org/10.1016/j.scs.2023.104486.

Zhou, Tao, Bo Huang, Rongrong Li, Xiaoqian Liu, and Zhihui Huang. 2022. "An Attention-Based Deep Learning Model for Citywide Traffic Flow Forecasting." *International Journal of Digital Earth* 15 (1). 323–344. https://doi.org/10.1080/17538947.2022.2028912.