Heliyon 9 (2023) e17746

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Dynamic graph neural network with adaptive edge attributes for air quality prediction: A case study in China

Jing Xu [a], Shuo Wang [a,e,d], Na Ying [b], Xiao Xiao [c], Jiang Zhang [a,e], Zhiling Jin [c], Yun Cheng [d], Gangfeng Zhang [f,g,*]

[a] School of Systems Science, Beijing Normal University, Beijing, 100875, China
[b] Chinese Research Academy of Environmental Sciences, Beijing, 100085, China
[c] School of Telecommunications Engineering, Xidian University, Xi'an, 710071, Shaanxi, China
[d] Information Technology and Electrical Engineering, ETH Zurich, Zurich, 8092, Switzerland
[e] Swarma Research, Beijing, China
[f] State Key Laboratory of Earth Surface Processes and Resource Ecology, Beijing Normal University, Beijing, 100875, China
[g] Faculty of Geophysical Science, Beijing Normal University, Beijing, 100875, China

ARTICLE INFO

Dataset link: https://climate.copernicus.eu/climate-reanalysis

Dataset link: https://english.mee.gov.cn/

Keywords:
Air quality prediction
Adaptive graph learning
Dynamic graph
Message passing neural networks

ABSTRACT

Air quality prediction is a typical Spatiotemporal modeling problem, which always uses different components to handle spatial and temporal dependencies in complex systems separately. Previous models based on time series analysis and recurrent neural network (RNN) methods have only modeled time series while ignoring spatial information. Previous graph convolution neural networks (GCNs) based methods usually require providing spatial correlation graph structure of observation sites in advance. The correlations among these sites and their strengths are usually calculated using prior information. However, due to the limitations of human cognition, limited prior information cannot reflect the real station-related structure or bring more effective information for accurate prediction. To this end, we propose a novel Dynamic Graph Neural Network with Adaptive Edge Attributes (DGN-AEA) on the message passing network, which generates the adaptive bidirected dynamic graph by learning the edge attributes as model parameters. Unlike prior information to establish edges, our method can obtain adaptive edge information through end-to-end training without any prior information. Thus reducing the complexity of the problem. Besides, the hidden structural information between the stations can be obtained as model by-products, which can help make some subsequent decision-making analyses. Experimental results show that our model received state-of-the-art performance than other baselines.

1. Introduction

Air quality has a significant impact on our daily lives. People who breathe clean air sleep better and are less likely to die prematurely from diseases such as cardiovascular and respiratory disorders, as well as lung cancer [1,2]. One of the key factors that decreases air quality is $PM_{2.5}$ (atmospheric particulate matter (PM) having a diameter of 2.5 μm or less), which can easily be

* Corresponding author at: State Key Laboratory of Earth Surface Processes and Resource Ecology, Beijing Normal University, Beijing, 100875, China.
E-mail address: zhanggf15@foxmail.com (G. Zhang).

https://doi.org/10.1016/j.heliyon.2023.e17746
Received 9 July 2022; Received in revised form 27 June 2023; Accepted 27 June 2023

Available online 3 July 2023

inhaled and cause damage to the human body. Thus, monitoring and forecasting the $PM_{2.5}$ concentrations are critical for improving air quality. With the rapid development of industry, a significant amount of energy is consumed, resulting in massive $PM_{2.5}$ emissions [3]. According to [4], in 2016, 38 days were heavily polluted due to $PM_{2.5}$ emissions in Beijing. High $PM_{2.5}$ concentrations may cause serious adverse health impacts and diseases [5], such as cardiac and pulmonary disease [6], detrimental effects on birth outcomes [7], and infant mortality [8]. Fortunately, to monitor and record air quality data, a large number of low-cost air quality sensors have been deployed, which makes it possible for researchers to perform accurate air quality prediction tasks. Accurate air quality predictions are useful, for example, individual activity arrangements and government pollution restrictions can benefit from that [9]. When the predicted $PM_{2.5}$ concentrations are too high, people can avoid going out and politicians can modify policies accordingly.

Conventional air quality prediction approaches can be generally divided into two categories: the knowledge-driven approach and the data-driven approach. In the past two decades, the knowledge-driven approach has been widely adopted for air quality prediction. The representations of this approach include the community multiscale air quality (CMAQ) [10], comprehensive air quality model with extensions (CAMx) [11], weather research, and forecasting/chemistry (WRF-Chem) [12] modeling systems. These kinds of knowledge-driven approaches strongly rely on the prior assumptions about the underlying physical and chemical processes involved in air pollution formation and transport. While these assumptions can provide a useful framework for understanding pollution dynamics, they are not always comprehensive and may not fully capture all relevant factors that contribute to air quality [13]. Additionally, determining the model parameters that govern these processes can be computationally intensive, particularly for complex models like CMAQ. This is because the model requires solving a set of differential equations that describe the behavior of various chemical species in the atmosphere over time. These equations involve many parameters that need to be estimated from observational data or laboratory experiments. In many cases, supercomputers are necessary to perform the large number of calculations required to estimate these parameters accurately. Even with the aid of supercomputers, the process can be time-consuming and computationally intensive. Due to these limitations, data-driven approaches such as machine learning techniques can help identify complex relationships and patterns in data with few prior assumptions about the underlying processes.

Recently, data-driven approaches have shown great performance in air quality prediction. As a conventional data-driven approach, statistical methods have been widely adopted for their simple structure. Yi et al. [9] applied the autoregressive integrated moving average (ARIMA) model to capture the trend of air quality time series in New Delhi. Naveen et al. [14] then adopted the seasonal autoregressive integrated moving average (SARIMA), which can capture the seasonal feature of time series, to predict the air quality in Kerala. However, due to the complexity and uncertainty of air quality prediction tasks, it is difficult for statistical methods to perform well for long-term predictions. Different from statistical methods, machine learning methods are non-parametric methods that can automatically learn new patterns and thus can handle the complex non-linearity of temporal data. In recent years, machine learning methods have been widely employed for air quality prediction, including the support vector regression (SVR) [15], the extreme gradient boosting (XGBoost) [16] algorithm and the random forest approach [17], etc. However, these methods do not take into account the spatiotemporal correlations and thus limiting their prediction performance.

To extract the spatiotemporal correlations, deep learning methods have been applied for air quality forecasting. Wen et al. [18] proposed a spatiotemporal convolutional long short-term memory (LSTM) neural network to capture the temporal and spatial dependencies. The temporal patterns were captured through the LSTM networks and the spatial dependencies were extracted by the convolutional neural networks (CNNs). Zhang et al. [19] then modeled the spatiotemporal correlations with the CNN model and the gated recurrent units (GRUs). The above methods can provide satisfactory prediction results, nevertheless, the CNN model is not suitable to model the non-Euclidean structure data and thus the spatial relationships between air sensors cannot be effectively modeled.

Most recently, graph-based deep learning methods have gained popularity since they can process the non-Euclidean structure data by modeling it to a graph for training [20,21]. Wang et al. [22] and Zhang et al. [19] separately employed graph convolutional networks (GCNs) to model the contextual relationships among air quality stations and further predict the air quality in the future. In a relatively short period, this modeling approach was very successful.

Models based on GCN need to construct the graph structure in advance. Traditional methods for constructing graph structures are usually based on prior knowledge, which can be divided into three categories: methods based on geographic distance, time-series similarity [22], and wind field information [23]. However, we cannot exhaustively enumerate all factors previously. Besides, parallel learning of too many graphs may result in too many parameters and high computational costs. In conclusion, inaccurate prior information may lead us to incorrectly connect two unrelated stations or lose links between two related stations. Moreover, the contextual relationships are constantly changing due to the impacts of the wind fields and other factors. Therefore, the dynamic graph is more suitable to model the relationships among stations in the real world [24,25]. But due to the incompleteness of prior knowledge, Wu et al. [26] proposed the model Graph WaveNet which developed an adaptive dependency matrix through node embeddings to capture hidden spatial dependency in data. But the node embeddings here are invariant on the time axis thus the graph learned is static. And this method also be used in Multivariate Time Series Forecasting with Graph Neural Networks (MTGNN) [27] to extract uni-directed relations among variables. However, the method of changing the adjacency matrix with epochs while training may introduce interference information and rise the difficulty of learning, which could potentially affect the prediction accuracy. As a result, Graph WaveNet did not perform well on actual air quality prediction datasets. To overcome these limitations, we develop a new method that learns the dynamic links between two stations automatically.

In this paper, we propose to construct a Dynamic Graph Neural Network with Adaptive Edge Attributes (DGN-AEA). Firstly, to address the shortcomings of prior information, we propose a method that uses self-use dynamic graph learning. However, the dynamic adjacency matrix represents the connection relationships between nodes will change with time. This kind of change will bring instability and difficulty to model training. So we divide the adjacency matrix into two parts, the connection relation (topology)

Fig. 1. The distribution of the observation station of the pollutant studied in our work. Each point represents one city-level observation station.

matrix, and the weight matrix, and propose to use an adaptive edge attributes (weights) matrix. Experiments show that the adaptive edge attributes can improve the prediction result. Secondly, in order to solve the physical consistency problem of many existing deep learning models, we designed a dynamic edge connection construction method using wind field information and combined it with adaptive edge connection through the method of multi-graph stitching. In this way, these learnable edges can be used as a correction of prior information, which can help the model get rid of the one-sidedness of prior knowledge. Thirdly, we also calculate the outbound and inbound directions respectively when aggregating the neighbor node's information. In this way, the inflow and outflow processes during the diffusion of pollutants are simulated. In summary, the contributions of this paper are listed as follows.

- We introduce the adaptive dynamic graph learning unit to learn the dual-path weighted edges automatically, to solve the problem of correlation graph modeling in non-Euclidean space.
- The wind field data can be integrated into our model as a type of directed dynamic connection by a Multi-Graph Process Block (MGP). The physical consistency of the model is improved in this way.
- For each node, we calculate its in-degree and out-degree separately to model convolution calculations on weighted directed graphs, which is more suitable for complex systems in the real world.
- The proposed DGN-AEA model improves the prediction capabilities and achieves state-of-the-art prediction accuracy.

The remainder of this paper is organized as follows. In Section 2, we introduce the method to construct the adaptive dynamic graph and our proposed DGN-AEA model. In Section 3, we describe the data used in our research and how we design experiments to verify the performance of DGN-AEA on the real-world air quality dataset. In Section 4, we show the results of experiments and try to discuss what makes DGN-AEA performs better. Finally, we conclude this work in Section 5 and discussed the disadvantages and future works in Section 6.

2. Method used

In this section, we first give the mathematical definition of air quality prediction. Next, we describe how we construct the two kinds of dynamic graphs. Then, as illustrated in Fig. 2, we introduce the DGN-AEA model which is designed to solve the adaptive graph learning problem. We show the details of how we leverage the framework of GCNs on the spatial domain to handle message passing on directed edges. We also use the spatial block Dynamic Multi-Graph Process Block (MGP) to combine the adapted edge attributes and the wind graph with MLPs, and the temporal block GRU. Finally, we form the stacked GCNs, which need spatial and temporal blocks working together to capture the spatiotemporal dependencies among cities.

2.1. Problem definition

Air quality prediction can be seen as a typical spatiotemporal prediction problem. Let $X^t \in R^N$ denote the observed $PM_{2.5}$ concentrations at time step t. The method based on GCNs usually models the changing spatial correlations among different cities by

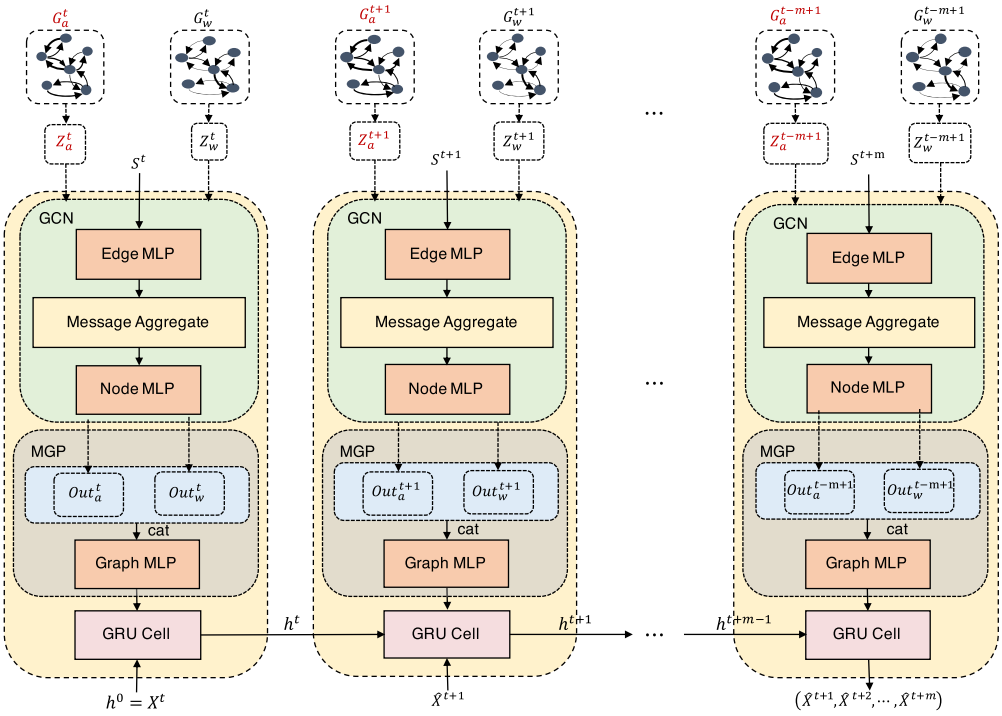


Fig. 2. Model structure of proposed model DGN-AEA.

the dynamic directed graph $G = (V, E, t)$, where V is the set of nodes and it is always the number of N. E^t is the set of weighted edges representing the potential interactions among cities where its weight may change over time. Let $S^t \in R^{N \times s}$ denote the nodes' attribute and $Z^t \in R^{N \times z}$ denote the edges' attribute at time step t, where s and z represent the variable dimensions of node features and edge features, respectively. The problem aims to predict the next T steps of $PM_{2.5}$ concentrations $[X^{t+1}, ..., X^{t+T}]$. based on the nodes' attribute $[S^{t+1}, ..., S^{t+T}]$ and the edges' attributes $[Z^{t+1}, ..., Z^{t+T}]$. The mapping among the input and output can be shown as follows:

$$\left[X^t; S^{t+1}, \cdots, S^{t+T}; Z^{t+1}, \cdots, Z^{t+T}\right] \xrightarrow{f(\cdot)} \left[\hat{X}^{t+1}, \cdots, \hat{X}^{t+T}\right] \tag{1}$$

where $\hat{X}^t$ represents the predicted vector, and $f(\cdot)$ is the prediction function based on the DGN-AEA framework.

2.2. Dynamic graph construction

In the air quality forecasting problem, we need to predict the future steps of all the cities. So the number of nodes will not change with time, which is different from some evolving dynamic graph problems [28–30].

We define the weighted adjacency matrix A, which can be divided into two parts: the topology matrix (P) of 0 or 1 indicating whether two nodes are connected, and the weight matrix (represents the edge attributes Z) indicating the strength of mutual influence between nodes. When using a neural network for training, if the adjacency matrix changes with time, it will bring great instability during training. Therefore, we use the adaptive edge attribute to represent this changing node interaction since it does not change the connection relationship between nodes but changes the strength of these connections, which means that the topology matrix will be static but the adjacency matrix will be dynamic. Thus, we suppose that all the correlations among cites are decided by the Euclidean distance (Equation (5)) like many previous graph-based air quality prediction approaches.

2.2.1. Topology and adjacency matrix

As we all know, the pollutant (such as $PM_{2.5}$, PM_{10}) concentrations in one place are strongly affected by other adjacents. Considering that relationships like that in the real world are usually sparse and different, to model these two spatial correlation features explicitly, we define the adjacency matrix A of Graph G in Equation (2):

$$A = P \odot Z, \tag{2}$$

where $\odot$ represents the Hadamard product, Z represents the edge attributes matrix. The formulation of Z will show in the following sections.

As to the topology matrix P, we introduce the effect of distance on-site relevance for the impact of the site is inversely proportional to the distance. And when the altitude between the two stations is too high, the connections will also be blocked out. To consider the above two factors, we use a Heaviside step function to filter out the edges that do not meet the rules as Equation (3):

$$P_{ij} = H\left(d - \xi_d\right) \cdot H\left(a - \xi_a\right), \tag{3}$$

where $H(\cdot)$ is the Heaviside step function see Equation (4):

$$H(x) = \begin{cases} 1, & \mathrm{x} > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{4}$$

where a represents the altitude of each node. d is the Euclidean distance calculated by the relative positions between two stations:

$$d = \sqrt{\left(x_1 - x_2\right)^2 + \left(y_1 - y_2\right)^2}. \tag{5}$$

Here the Euclidean distance and altitude threshold ξ_d and ξ_a equals 300 km and 1.2 km respectively. Here we get the topology matrix P.

2.2.2. Node attributes

The node attributes S^t are mainly meteorological data. Which includes 17 types of variables same as [23]. We chose 8 of them as the final node attributes: *Temperature*, *Planetary Boundary Layer height*, *K index*, *Relative humidity*, *Surface pressure*, *Total precipitation* and *the u and v component of wind*. The time interval is consistent with these node attributes with the $PM_{2.5}$ concentration data as 3 h.

2.2.3. Edge attributes

In our work, there are two kinds of edge attributes: One is from the wind field and another is from the adaptive neural network parameter. We use the advection coefficient as attributes from wind data and calculate it as Equation (6):

$$Z_w^t = relu\left(\frac{\left|\vec{v}^t\right|}{d} \cdot \cos\left(\alpha - \beta\right)\right), \tag{6}$$

where $\vec{v}^t$ represents the wind speed at time t, d is the distance between stations, and α and β are angles of cities and wind directions. $relu(\cdot)$ is the ReLU activation function.

$Z_a^t \in R^{1\times l}$ the adaptive neural network parameters, where l represents the number of edges, i.e., the number of 1 in the topology matrix P. We set it as one important parameter which can be seen as another kind of useful edge attribute in addition to wind effects in the air quality prediction problem. This parameter can be obtained by continuous iterative optimization through the training stage.

By setting adaptive dynamic edge weights as learnable parameters, such dynamic correlations can be directly learned during the end-to-end training process. Even in practical scenarios where some prior information is missing, the correlation network between sites can still be adaptively learned for spatiotemporal prediction. When using wind field information, we can consider this learnable parameter as a supplement to wind field information. The prediction accuracy can be further improved in this way. The details will be presented in Section 3.

2.3. Dynamic graph neural network with adaptive edge attributes

2.3.1. Graph convolution block

Many dynamic graph neural network methods are based on the spectral domain. The convolution operation on the graph is equivalent to the product in the spectral domain after the Fourier transform. The corresponding Fourier transform basis is the eigenvector of the Laplace matrix. And the model Chebnet [31] uses the Chebyshev polynomial to approximate the spectral convolution. However, these methods cannot handle the directed graph since the Laplacian matrices are used for undirected graphs [32,33]. They are not suitable for complex system modeling because many relations in complex systems are directed. Besides, the prediction accuracy is limited by the order of the Chebyshev polynomial fit, and in many cases does not perform as well as spatial GCNs [34]. To solve these problems, we take the spatial domain GCN i.e., the Message Passing Neural Network (MPNN) in use.

MPNN framework can be divided into two stages: message passing stage and readout stage [35]. Compared with spectral-domain GCN which can only model node attributes, MPNN directly aggregates messages from neighbor nodes and can also model edges, which makes it more flexible and intuitive. For node i at time t, our GCN block with MPNN framework will work as the following equations:

$$\varepsilon_i^t = \left[\hat{X}_i^{t-1}, S_i^t\right], \tag{7}$$

$$m_{ij}^t = \varphi\left(\left[\varepsilon_i^t, \varepsilon_j^t, Z_{ij}^t\right]\right), \tag{8}$$

$$e_i^t = \omega\left(\sum_{j\in N(i)}\left(m_{ij}^t - m_{ji}^t\right)\right), \tag{9}$$

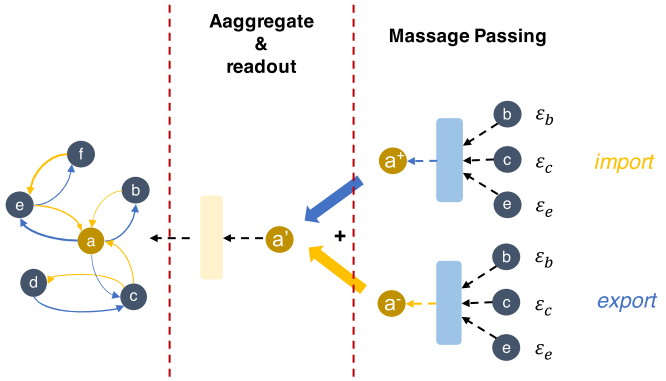


Fig. 3. Illustration for our used MPNN to process incoming and outgoing edge processing separately (use node a as an example). Different colors of arrows represent the import and export process separately.

where $[\cdot,\cdot]$ represents the concat operator that merges two 1-D vectors into a single vector. Equation (7) represents the splicing operation of neighbor information and edge connection weights. ε_i^t in Equation (7) represents the result after the concatenation operation of the input matrix. $\varphi(\cdot)$ in Equation (8) represents one layer of MLP. $N(i)$ represents the neighbors of node i. The whole Equation (8) represents the process of aggregating neighbor information according to the edge weights. m_{ij} and m_{ji} respectively represent the in-degree and out-degree information of the node. In Equation (9), $\omega(\cdot)$ is another layer of MLP. After Equation (9), we can calculate the increase and decrease after the node's message passing process.

In our proposed method, we use two kinds of edge attributes. For each different edge attribute, Z may represent Z_w or Z_a, then we can get e_w and e_a following Equation (9). Next, we concat the two graph-level embeddings through the transfer layer:

$$\zeta^t = \psi\left([e_w^t, e_a^t]\right). \tag{10}$$

It should be noted that Z_a is a learnable parameter in our model, thus it will be updated through the model training stage. After that, we can get an adaptive edge attribute Z_{AEA}.

Since the $PM_{2.5}$ transport graph network is a directed graph, in order to realize the material conservation of source and sink nodes, we calculate the message aggregation process of each node's incoming and outgoing edges respectively. This is consistent with the physical process of pollutant diffusion. It can improve the prediction accuracy of the model. The specific calculation process is shown in the following Fig. 3.

2.3.2. *Temporal processing block*

Our model DGN-AEA in Fig. 2 can be seen as a stacked structure: it is processed by MPNN, which is good at processing information on graphs, and then input the aggregated results to the GRU, which is good at iterative prediction of time series. Such a structure can more clearly extract effective information in the temporal and spatial domains, respectively. Here we will introduce the temporal processing block GRU. The input of GRU includes the historical $PM_{2.5}$ concentrations of the nodes and future meteorological data, which is embedded in the node embedding ϵ_i^t. We also input the information gathered on the graph to the GRU. The prediction process for node i at time t is shown as follows:

$$c_i^t = \left[\zeta_i^t, \epsilon_i^t\right]. \tag{11}$$

This concatenates the input variables (results after graph convolution and the meteorological variables) to prepare for subsequent matrix operations:

$$q_i^t = \sigma\left(W_q \cdot \left[h_i^{t-1}, c_i^t\right]\right), \tag{12}$$

$$r_i^t = \sigma\left(W_r \cdot \left[h_i^{t-1}, c_i^t\right]\right), \tag{13}$$

where q_i^t and r_i^t in Equation (12) represent the results after operations of the update gate and forget gate in GRU respectively. $\sigma(\cdot)$ represents the sigmoid activate function. h_i^{t-1} is the hidden state of the previous time step, c_i^t represents the aggregated input features after Equation (11). The update gate is used to control the degree to which the state information of the previous moment is brought into the current state. The state information brought in at the previous moment is somewhat positively related to the value of the update gate. The reset gate controls how much information from the previous state is written to the current candidate set $\widetilde{h}_i^t$. The smaller the reset gate, the less information from the previous state is written. $\widetilde{h}_i^t$ is calculated by:

$$\widetilde{h}_i^t = tanh\left(\widetilde{W} \cdot \left[r_i^t * h_i^{t-1}, c_i^t\right]\right), \tag{14}$$

where W_q, W_r, $\widetilde{W}$ in Equation (12) to Equation (14) are learnable parameters. After the gate control signal is obtained, we first use the reset gate to obtain the data after "reset". After the Hadamard product operation of the reset gate r_i^t and the hidden layer

Algorithm 1 $PM_{2.5}$ Prediction Algorithm.

Input: Historical $PM_{2.5}$ concentrations X^0;
Node's attributes $S = [S_1, \cdots, S_T]$;
Edge's attributes(by wind) $Z_w = [Z_w^1, \cdots, Z_w^T]$;
Randomly initialized adaptive edge's attributes $Z_a = [Z_a^1, \cdots, Z_a^T]$;
Output: Future $PM_{2.5}$ concentrations $\hat{X} = [\hat{X}^1, \ldots, \hat{X}^{1+T}]$;
Learned adaptive edge attributes $Z_{AEA} = [Z_{AEA}^1, \ldots, Z_{AEA}^{1+T}]$;
Evaluation metric MAE and RMSE.

$\hat{X}^0 = X^0$
$h^0 = 0$
for each training epoch $epoch \in [1, 50]$ **do**
  for each time step $t \in [1, T]$ **do**
    for all $v_i \in \mathbf{V}$ **do**
      $\zeta_i^t = \psi\left(MPNN1\left(\hat{X}_i^0, S_i^t, Z_w^t\right), MPNN2\left(\hat{X}_i^0, S_i^t, Z_a^t\right)\right)$;
      $\hat{X}_i^t = GRU(\varepsilon_i^t, \zeta_i^t, h_i^{t-1})$;
      $\hat{X} = [\hat{X}, \hat{X}_i^t]$.
    end for
  end for
  Compute loss between $\hat{X}$ and X use Equation (17).
  Gradient update the adaptive edge attributes Z_{AEA}, and other learnable parameters such as W_q, W_r, $\widetilde{W}$.
end for
Calculate MAE and RMSE follow the Equation (18) and Equation (19);
return $\hat{X}$, Z_{AEA}, MAE, RMSE

information of the previous step h_i^{t-1}, then spliced the input signal c_i^t of the current step. Then we multiply the result of the above operation by a learnable matrix and then let the result pass a *tanh* activation function to scale the result to the range of [-1, 1]. As shown in Equation (14). The input information is added to the current hidden state in a targeted manner, which is equivalent to memorizing the state at the current moment.

Finally, Equation (15) performs the memory and updates operations at the same time, and obtains the updated hidden layer state:

$$h_i^t = \left(1 - q_i^t\right) * h_i^{t-1} + q_i^t * \widetilde{h}_i^t, \tag{15}$$

After the whole operation, we finally get the prediction result of $PM_{2.5}$ concentration by Equation (16):

$$\hat{X}_i^t = \Omega\left(h_i^t\right), \tag{16}$$

where $\Omega(\cdot)$ is a MLP layer.

2.3.3. Proposed learning algorithm

We use the stacked spatiotemporal prediction structure. As shown in Fig. 2, we first use MPNN to perform a convolution operation on the $PM_{2.5}$ concentration of each site on the graph according to the edge weight. Two MPNN blocks are used to process the wind field information map and adaptive dynamic map respectively. Then the two graph processing results are aggregated through a layer of MLP (as shown in Equation (7) to Equation (10)). Then, the output of the entire graph convolution part and the historical meteorological data [36,23] are input into the GRU for time series iterative processing as Equation (11) to Equation (15). Finally, we get the future forecast result as Equation (15). The whole process of the proposed DGN-AEA is shown in Algorithm 1.

3. Data and experiment design

In this section, we will show the details of our selected dataset and experiment settings.

3.1. Experiment settings

Our experiments are conducted on a Linux system with CPU: Intel(R) Xeon(R) Gold 5218 CPU @ 2.30 GHz and GPU: NVIDIA Corporation Device 2204 (rev a1) The batch size of model training, validation, and test data are all 32. All models are trained up to 50 epochs by early stop rules with 10 steps and use RMSProp optimizer. The learning rate is 5e-4 and weight decay is also set to 5e-4. All the prediction results are the average result after 10 repetitions. In the training stage, we aim to minimize the Mean Square Error (MSE) Loss function as the following equation:

$$MSE\ Loss = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{N}\sum_{i=1}^{N}\left(\hat{X}_i^t - X_i^t\right)^2\right), \tag{17}$$

where T is the length of the prediction time step, and N represents the number of samples. $\hat{X}$ and X represent the predicted value and the ground truth of $PM_{2.5}$ concentrations respectively.

To evaluate the prediction accuracy between models, we adopt two evaluation metrics: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

$$MAE = \frac{1}{N} \times \sum_{i=1}^{N} |\hat{y}_i - y_i|, \tag{18}$$

$$RMSE = \sqrt{\frac{1}{N} \times \sum_{i=1}^{N} (\hat{y}_i - y_i)^2}, \tag{19}$$

where y is ground truth and $\hat{y}$ represents the prediction results given by models. These two are commonly used indicators to evaluate the accuracy of time series forecasting.

3.2. Data used

To examine the ability of the model to solve real problems, we conduct experiments on the real-world datasets from the previous work [9]. This dataset is collected from MEE[1] and ERA5[2]. It contains three types of data: Sites' geographic information, meteorological data, and pollutant concentration ($PM_{2.5}$) data. The last two types of variables are time series data ranging from 2015-1-1 00:00:00 to 2018-12-31 23:59:59, with 3 hours for each time step. The dataset includes 184 city-level observation stations as shown in previous Fig. 1.

To test the predictive ability of the model under different circumstances. We divide the dataset into three parts by time. The training set of the first dataset is the data for two years in 2015 and 2016, and the test set and validation set are the data for the whole year of 2017 and 2018, respectively. The training, validation, and testing of data set 2 were sequentially intercepted in the winter of 2015 - 2018 for three consecutive years (November 1st - February 28th of the following year). This is because winter is usually the season of high $PM_{2.5}$ pollution in China, and the average value of the data is higher. Dataset 3 uses the 2016 autumn and winter 4 months (September 1st - November 30th) for training and uses the winter data of the following two months for verification (December 1st to December 31st) and test (January 1st to January 31st of the following year) respectively. The period of 2016 was chosen because that winter saw almost the worst pollution in Chinese history.

The amount of training data and time periods differ among the three datasets. Dataset 1 includes 2 years of time series data for training, while dataset 2 and 3 only have 4 and 3 months, respectively. Generally, more data can enhance the model's training. Moreover, the time period of dataset 3 coincides with the frequent occurrence of heavy pollution in China, resulting in a higher mean value. This enables testing the model's ability to predict pollutant concentrations during heavy pollution seasons. It is worth noting that the experimental settings for all three datasets are the same.

3.3. Baselines

In our work, we consider baselines to examine the model effect. Baselines include classical statistical models, classical spatiotemporal prediction models, and state-of-the-art deep learning models with adaptive graph components.

- **HA:** The Historical Average (HA) model is a typical time series analysis model, which main idea is to use the average of all the values at the corresponding time in history (known data) as the predicted value for the future. Therefore, there is no concept of the prediction time step. Here we refer to the construction method in the article [37] to calculate the test dataset and intercept all the moments of one week to predict the corresponding time points.
- **LSTM:** The long short-term memory (LSTM) [38] model is an improvement of the RNN model, which uses three types of gates to extract more useful related historical data.
- **GC-LSTM:** GC-LSTM [39] is a model which uses two spectral-based GCNs embedded into the long-short term memory model to extract spatiotemporal features from data.
- $PM_{2.5}$**-GNN:** $PM_{2.5}$-GNN [23] is a state-of-the-art prediction model for $PM_{2.5}$ concentrations prediction. It also uses the stacked spatiotemporal structure based on GCNs and RNNs.
- **Graph WaveNet (w/o weather):** The Graph WaveNet [26] develops a novel adaptive dependency matrix, which can automatically capture the spatial dependency from data. It uses Temporal Convolutional Network (TCN) as the temporal block. It has achieved state-of-the-art results in many real-world datasets, especially in traffic flow forecasts. Since the original model uses one-dimensional convolution to operate on only one variable, we do not use multi-dimensional meteorological information when reproducing the model. The results will be presented in Fig. 6, and a detailed comparison with DGN-AEA (w/o weather) can be found in Table E.3.

3.4. Ablation study

In order to further illustrate the role of the adaptive dynamic edge attribute, we compare the results with models with some parts removed, which are:

[1] https://english.mee.gov.cn/.
[2] https://climate.copernicus.eu/climate- reanalysis.

Table 1
Prediction accuracy compared with baselines.

Dataset	Methods / Metric		HA	LSTM	GC-LSTM	$PM_{2.5}$-GNN	DGN-AEA
Dataset 1	RMSE	3	25.81	12.17±0.10	12.03±0.07	11.55±0.11	**11.34±0.07**
		6		15.61±0.11	15.40±0.07	14.76±0.10	**14.53±0.09**
		12		18.56±0.11	18.38±0.09	17.70±0.15	**17.06±0.11**
		24		20.87±0.16	20.81±0.09	20.18±0.17	**19.20±0.15**
	MAE	3	37.26	9.43±0.09	9.31±0.06	8.93±0.05	**8.75±0.05**
		6		12.44±0.12	12.25±0.06	11.70±0.10	**11.50±0.08**
		12		14.88±0.14	14.69±0.10	14.11±0.17	**13.56±0.11**
		24		16.48±0.15	16.45±0.08	15.90±0.19	**15.06±0.14**
Dataset 2	RMSE	3	52.21	18.01±0.17	18.30±0.11	17.61±0.17	**17.15±0.07**
		6		23.55±0.22	23.65±0.19	22.94±0.24	**22.29±0.09**
		12		28.60±0.23	28.547±0.17	27.52±0.30	**26.85±0.11**
		24		32.82±0.27	33.03±0.33	31.70±0.29	**30.79±0.20**
	MAE	3	35.84	14.01±0.15	14.21±0.08	13.70±0.15	**13.33±0.06**
		6		18.90±0.19	18.95±0.19	18.38±0.23	**17.83±0.09**
		12		23.14±0.21	22.96±0.16	25.26±0.41	**21.60±0.10**
		24		26.61±0.32	26.37±0.36	25.26±0.41	**24.45±0.22**
Dataset 3	RMSE	3	42.33	26.43±0.36	26.56±0.20	25.51±0.32	**24.75±0.17**
		6		33.87±0.41	34.06±0.27	32.95±0.33	**31.98±0.36**
		12		40.98±0.55	40.84±0.66	39.76±0.71	**38.78±0.27**
		24		45.08±1.02	44.86±0.70	45.04±0.88	**42.18±0.84**
	MAE	3	29.31	20.52±0.28	20.63±0.18	19.84±0.27	**19.22±0.14**
		6		27.14±0.40	27.23±0.23	26.37±0.32	**25.53±0.34**
		12		33.16±0.59	32.92±0.58	32.14±0.74	**31.19±0.24**
		24		36.89±1.01	36.62±0.73	36.23±0.99	**34.08±0.78**

- **Static:** To demonstrate the role of dynamic graphs, we conduct experiments using only static graph structures based on distance and altitude calculations.
- **Only AEA:** As described before, DGN-AEA integrates wind edge information. Here the wind information is removed and only adaptive edge attributes are used. This can illustrate the important role of wind in modeling $PM_{2.5}$ forecasting.
- **Only Wind:** Contrary to **Only AEA**, here we only use the wind field information and remove the adaptive edge attributes. Similar to the control variables approach, this can illustrate the importance of using adaptive edge attributes.
- **W/O weather:** Here we also compare the effect of not inputting the future weather GRU module as known information to illustrate the effect of using future weather.
- **AEA+Wind(ours):** As shown in Fig. 2, we will use the multi-graph information of adaptive edge attributes and wind at the same time.

4. Results and discussion

4.1. Performance comparisons with baselines

We compare the prediction results with the evaluation metric between our DGN-AEA and baselines in all three datasets. In addition, we set different prediction horizon time steps with 3 (9 h), 6 (18 h), 12 (36 h), and 24 (72 h) so that we can compare the predictive ability of various models under different time prediction lengths. The best results are highlighted in boldface in Table 1.

We divided the data into three datasets. By designing the number of samples and seasons of the training set dataset, it can be considered that the training difficulty on the three datasets is increasing. The training set of dataset 3 has the least data and the corresponding value is large, since winter is the high season of haze in China, the value of $PM_{2.5}$ is generally higher. All the results in Table 1 have been repeated 10 times without fixed random seeds, so parameter sensitivity can be ruled out.

It can be seen that our model always performs the best and the traditional statistical model HA is not always the worst. GC-LSTM performs a little better than LSTM, nevertheless, it does not perform well on our dataset overall. In dataset 3, our model DGN-AEA improves the RMSE of GC-LSTM by 6.81%, 6.11%, 5.04%, and 6.7%, respectively. $PM_{2.5}$-GNN performs better than other baselines, but our model is more accurate. On the RMSE of dataset 3, DGN-AEA is 2.98%, 2.94%, 2.46%, 6.3% more accurate than $PM_{2.5}$-GNN.

Compared with another adaptive graph model, Graph WaveNet requires a large number of parameters and has a high computational resource overhead, so the training is slow. Its effect is also not good. Compared with Graph WaveNet, our proposed model DGN-AEA improves the RMSE of dataset 3 by 47.46%, 45.32%, 44.56%, 43.98%, and 38.08%, 36.05%, 37.61%, 36.99% on MAE (shown in Fig. 6 and Table E.3). We speculate that it may be due to the construction of the adjacency matrix that changes from time

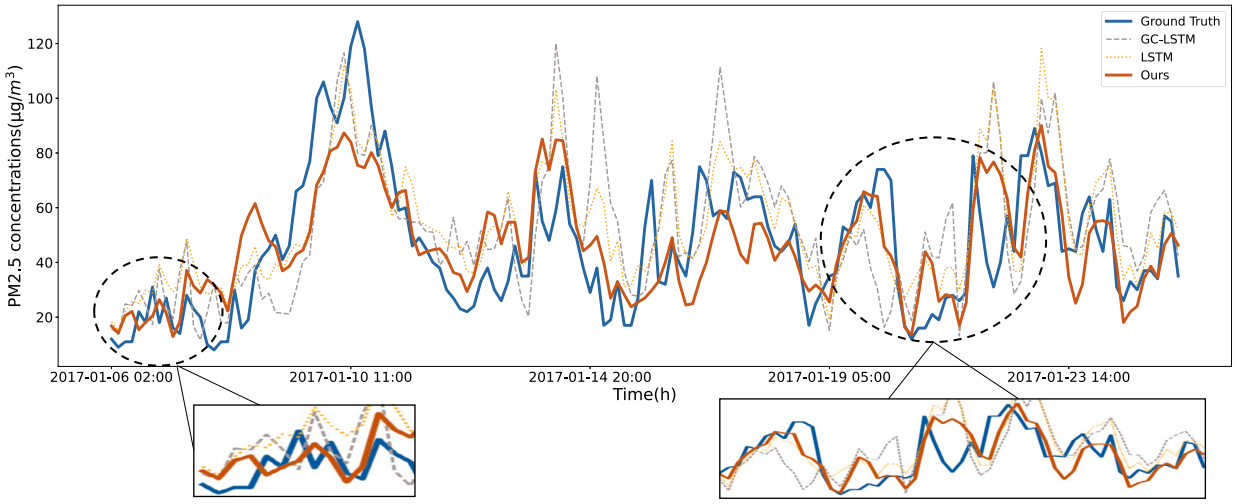


Fig. 4. Future $PM_{2.5}$ concentrations predicted by DGN-AEA.

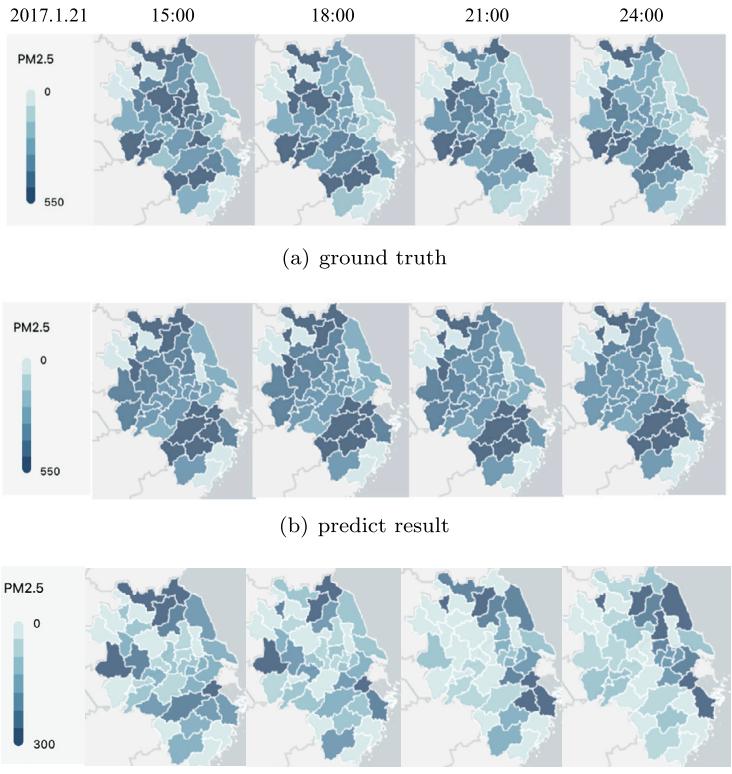


Fig. 5. Visualization of forecast results in the Yangtze River Delta region. The data is derived from dataset 3 with 24 prediction horizon time steps. (The unit is $\mu g/m^3$).

to time and brings great difficulty to training on $PM_{2.5}$ datasets, making it difficult for the model to grasp the exact topology of the stations.

The prediction fit curves of Linan are also plotted in Fig. 4. We achieve the most accurate result compared with other models.

The above results are the average of the whole map and fit curves for individual cities. In order to examine the prediction ability of the model at the local regional scale, we select the ground truths and predicted results of the Yangtze River Delta region for a continuous period of time to visualize. The result is shown in Fig. 5.

Table 2
Result of ablation study.

Dataset	Methods / Metric		Only Wind	Only AEA	Static	W/O weather	AEA + Wind
Dataset 1	RMSE	3	11.55±0.05	11.38±0.04	11.58±0.08	13.03±0.06	**11.34±0.06**
		6	14.76±0.10	14.55±0.06	14.72±0.09	17.08±0.07	**14.53±0.09**
		12	17.70±0.15	17.36±0.07	17.51±0.13	20.78±0.10	**17.06±0.11**
		24	20.18±0.17	19.78±0.09	19.80±0.19	24.41±0.08	**19.20±0.15**
	MAE	3	8.93±0.05	8.78±0.04	8.95±0.06	10.14±0.05	**8.75±0.05**
		6	11.70±0.10	11.51±0.06	11.68±0.08	13.73±0.06	**11.50±0.08**
		12	14.11±0.17	13.79±0.08	13.97±0.13	16.90±0.11	**13.56±0.11**
		24	15.90±0.19	15.55±0.09	15.60±0.19	18.84±0.09	**15.06±0.14**
Dataset 2	RMSE	3	17.61±0.17	17.60±0.14	17.35±0.16	19.62±0.04	**17.15±0.07**
		6	22.99±0.21	23.02±0.20	22.34±0.08	26.17±0.03	**22.29±0.09**
		12	27.60±0.30	27.71±0.25	27.15±0.13	32.87±0.07	**26.85±0.11**
		24	31.70±0.29	31.50±0.33	31.49±0.34	38.89±0.08	**30.79±0.20**
	MAE	3	13.70±0.15	13.69±0.13	13.69±0.13	15.31±0.04	**13.33±0.06**
		6	18.41±0.20	18.45±0.20	18.67±0.08	21.12±0.03	**17.83±0.09**
		12	22.26±0.32	22.37±0.23	22.35±0.14	26.99±0.08	**21.60±0.10**
		24	25.26±0.41	25.06±0.31	24.81±0.33	31.94±0.11	**24.45±0.22**
Dataset 3	RMSE	3	25.51±0.32	25.32±0.23	24.80±0.28	27.16±0.05	**24.75±0.17**
		9	32.95±0.33	32.74±0.46	32.13±0.22	36.12±0.14	**31.98±0.36**
		12	39.10±0.63	39.26±0.26	39.31±0.41	44.91±0.13	**38.78±0.27**
		24	43.44±0.42	42.83±0.52	42.32±0.77	49.71±0.11	**42.18±0.84**
	MAE	3	19.84±0.27	19.68±0.20	19.26±0.24	21.15±0.05	**19.22±0.14**
		6	26.37±0.32	26.17±0.45	25.63±0.22	29.11±0.14	**25.53±0.34**
		12	31.54±0.64	31.51±0.28	31.72±0.42	36.93±0.16	**31.19±0.24**
		24	35.72±0.45	35.08±0.53	34.58±0.79	41.44±0.13	**34.08±0.78**

4.2. The prediction accuracy

As shown in Table 1, it can be observed that as the prediction horizon increases, the difference between predicted and observed data also increases. However, our proposed DGN-AEA model consistently holds the lowest prediction error. Furthermore, in Fig. 5, which displays the prediction results (Fig. 5(b)), ground truth (Fig. 5(a)) and the MAE value between the two (Fig. 5(c)) on four randomly selected time steps, we can see that our proposed model achieves accurate predictions across different locations.

4.3. The function of dynamic graph

To explain why we choose to retain two types of graph edges, we conduct ablation studies to explore the predictive effects of using different edges. We find that our proposed model is the best in most cases, regardless of the dataset or prediction at any time scale. Results in Table 2 also have been repeated 10 times without fixed random seeds, so parameter sensitivity can be ruled out.

We can see that the models where we do not use dynamic information perform worse than the dynamic graph method in the vast majority of cases.

In addition, the prediction accuracy of *Only Wind* and *Only AEA* are similar, and in most cases *Only AEA* has even better MAE and RMSE metrics. Especially in Dataset 3, our proposed DGN-AEA has an average of 4.3%,4.7%, 3.4%, and 7.8% MAE metric decrease than the other two dynamic edge attribute.

4.4. The function of future weather

Our model uses future weather data known for future air quality prediction. We compare DGN-AEA and DGN-AEA (w/o weather) models respectively. As shown in Fig. 6, by using future meteorological data, the prediction accuracy can be improved by 9.13%, 12.30%, 15.54%, 17.76% on MAE (Fig. 6(a) to Fig. 6(b)) and 8.87%, 11.76%, 13.65%, 15.15% on RMSE (Fig. 6(e) to Fig. 6(f)). This shows that it is very useful to use future weather data.

We compared our results with another adaptive graph model without future weather, Graph WaveNet. Graph WaveNet requires a large number of parameters and has a high computational resource overhead, so the training is slow. What's more, because the TCN model used by the time series module of Graph WaveNet can only perform convolution operations on one-dimensional variables, it cannot input future weather data. Compared with Graph WaveNet, our proposed model DGN-AEA(w/o weather) improves the RMSE of dataset 3 by 42.35%, 38.25%, 35.80%, 33.98%, and 31.86%, 27.10%, 26.13%, 23.39% on MAE. We speculate that it may be due to the construction of the adjacency matrix that changes from time to time and brings great difficulty to training on $PM_{2.5}$ datasets, making it difficult for the model to grasp the exact topology of the stations.

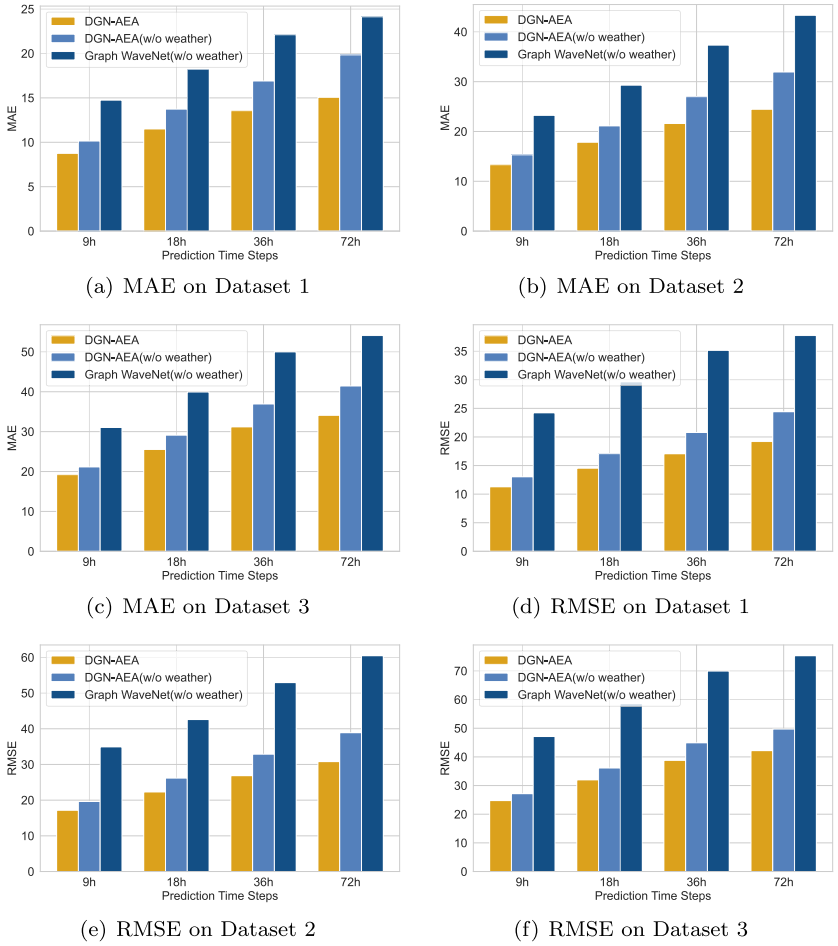


Fig. 6. Comparison among DGN-AEA, DGN-AEA (w/o weather) and Graph WaveNet(w/o weather) models. The results are the average of ten training sessions.

4.5. Comparison of two attributes

Following the above steps, we can get two kinds of graph edge attributes: one can be calculated by wind field data (Fig. 7(b)), and another can be learned after training (Fig. 7(a)). To get through the difference (Fig. 7(c)) and explain why the adaptive edge attribute is useful, we visualize both of them at the same time step shown in Fig. 7.

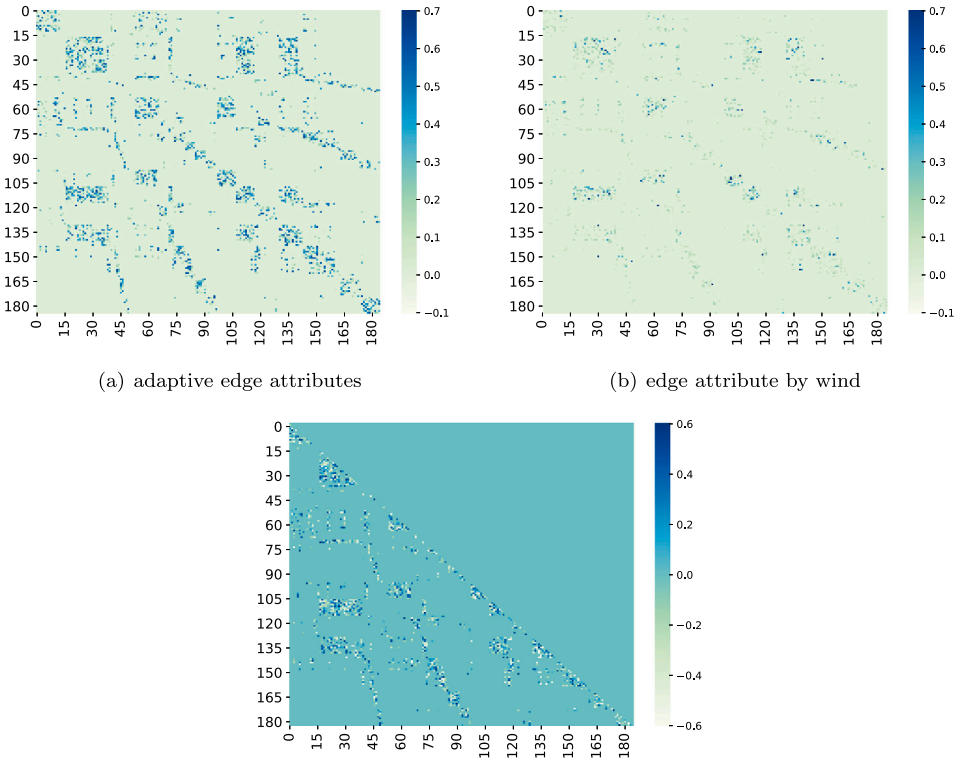
4.6. Complex network analysis on the learned adaptive edges

With DGN-AEA, an adaptive correlation network structure can be obtained after the training phase. The properties of the obtained correlation network can be discussed with the help of some analytical methods and indicators in the field of complex networks. We listed it in the supplementary information.

We separately count the sum of the weights on the incoming and outgoing edges of different nodes and calculate the total weight of the connected edges minus the total weight of the incoming and outgoing edges. The positive or negative value of this difference indicates that the node belongs to the type that is more affected by the surroundings or has a greater influence on the surrounding (Fig. E.8). At the same time, we can also compare the relationship between edge weights and degree (Equation (E.1)), and the relationship between edge weights and the degree centrality (Equation (E.2)) of the complex network. Shown in Fig. E.9 and Fig. E.10. We see a clear positive correlation between node weight and degree value. However, there is no obvious correlation between node degree centrality and degree value.

5. Conclusions

In this paper, we propose flexible Dynamic Graph Neural Networks with Adaptive Edge Attributes (DGN-AEA) based on the spatial domain. This method retains edges by the wind to follow the basic prior physical knowledge of air pollution transmission. At the same time, we calculate the transmission volume on the outgoing edge and incoming edge respectively when doing message transmission



(a) adaptive edge attributes

(b) edge attribute by wind

(c) The transpose of the upper triangle matrix of the adaptive edge attributes matrix minus the value of the lower triangle.

Fig. 7. Difference between the two matrices used in our model.

and aggregation to the nodes, which simulates the law of conservation of matter in the transport and diffusion of pollutants to some extent. Besides, we fuse adaptive edge attributes using the multi-graph structure. Experiment results show that our model achieves a better level of prediction effect on the real-world $PM_{2.5}$ dataset.

In this way, we can adaptively learn the correlation between real sites and obtain better time series prediction results. However, how much of the network relationship in the real world can be restored in real data sets by this way of adaptively constructing learnable parameters is also worth exploring. There are still some ideas of network reconstruction methods that may be worth learning from.

6. Limitation and future work

This work assumes that future air quality prediction results should be derived from weather forecast data. However, due to the time delay of ERA5 forecast reanalysis data, the use of different data during the training and prediction phases may lead to increased model errors. Hence, future studies should explore the possibility of incorporating weather forecast data to address this issue.

Additionally, a limitation of the model discussed in this paper is its reliance on a parameter matrix for constructing learnable edges. The process of learning this matrix is a black box, posing challenges for conducting a comprehensive principle analysis. Moreover, it is essential to note that our study utilized 184 city-level stations in China. However, it is worth mentioning that the distribution of these stations is not evenly spread throughout the country, and the potential heavy pollution issues in certain areas outside our sample have not been considered.

Furthermore, it was observed that the model occasionally exhibits inadequate prediction accuracy for extremely heavy pollution events. Consequently, there exists significant potential for enhancing spatiotemporal modeling. For example, incorporating prior information, utilizing high-order graph neural networks to model the pollutant reaction process, and integrating more effective adaptive components are prospective areas for improvement.

CRediT authorship contribution statement

Jing Xu: conceived and designed the experiments; performed the experiments; analyzed and interpreted the data; wrote the paper; **Shuo Wang:** conceived and designed the experiments; wrote the paper; **Na Ying:** contributed reagents, materials, analysis tools or data; **Xiao Xiao:** performed the experiments; wrote the paper; **Jiang Zhang:** wrote the paper; **Zhiling Jin:** analyzed and

interpreted the data; wrote the paper; **Yun Cheng:** wrote the paper; **Gangfeng Zhang:** conceived and designed the experiments; contributed reagents, materials, analysis tools or data; wrote the paper.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Jing Xu has patent pending to Jing Xu, Shuo Wang, Na Ying, Zhiling Jin, Yun Cheng, Jiang Zhang.

Data availability

Data associated with this study has been deposited at ERA5: https://climate.copernicus.eu/climate-reanalysis and MEE: https://english.mee.gov.cn/.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (NO. 42101027). We thank the support from the Save 2050 Program jointly sponsored by Swarma Club and X-Order. Shuo Wang acknowledges the financial support by China Scholarship Council (CSC) Grant No. 202106040117. Gangfeng Zhang acknowledges the financial support by the Second Tibetan Plateau Scientific Expedition and Research Program (STEP) Grant No. 2019QZKK0606 and Qinghai Province Key research and development and transformation project (2022-SF-173).

The authors wish to acknowledge the Editor and the anonymous reviewers for their detailed and helpful comments on the original manuscript.

Appendix A. Related work about air quality prediction

Air quality prediction issues have been studied for years. These issues were first studied by some conventional statistical methods, e.g., autoregressive integrated moving average (ARIMA) [9,40]. However, there are too much uncertainty and non-linearity in air quality prediction, which is not suitable for these statistical models to achieve high prediction accuracy for long-term prediction.

Machine learning methods make use of historical observations to perform accurate predictions. Liu et al. [15] proposed a multi-dimensional collaborative support vector regression (SVR) model for air quality index (AQI) forecasting in the Beijing-Tianjin-Hebei region while considering the weather conditions. Dun et al. [41] adopted the linear regression (LR) and SVR methods for short-term air quality prediction. Liu et al. [42] fused the principal component regression (PCR), SVR, and autoregressive moving average (ARMA) models to predict air quality with six different kinds of pollutants. However, these machine learning methods did not capture the spatiotemporal correlations and thus limited the prediction performance.

In recent years, deep learning methods are widely employed in air quality prediction issues due to their high prediction accuracy. Ma et al. [43] propose a transfer learning-based stacked bidirectional long short-term memory (LSTM) model which combined deep learning and transfer learning strategies to predict the air quality of some stations based on the data observed by other stations. Wen et al. [18] proposed a spatiotemporal convolutional long short-term memory neural network to capture the temporal and spatial dependencies with LSTM and convolutional neural networks (CNNs), respectively. Zhang et al. [19] proposed a hybrid model (MTD-CNN-GRU) for $PM_{2.5}$ concentration prediction. In the MTD-CNN-GRU model, the CNNs were employed to extract the spatial relationships, and the gated recurrent units (GRUs) were applied to capture temporal features. In this way, they could capture the spatiotemporal correlations to achieve higher prediction accuracy.

Appendix B. Related work about graph-based prediction methods

Conventional deep learning methods are not suitable for data processing in non-Euclidean space, which can not model the spatial correlations very well. To solve the problem, graph-based deep learning methods are proposed and have been widely applied to air quality forecasting these years. Wang et al. [22] proposed an Attentive Temporal Graph Convolutional Network (ATGCN) for air quality prediction. The ATGCN encoded three types of relationships among air quality stations including spatial adjacency, functional similarity, and temporal pattern similarity into graphs and aggregated features using gated recurrent units (GRUs). Finally, a decoder was designed to conduct multi-step predictions. Qi et al. [39] then proposed a GC-LSTM model which combined the graph convolutional networks (GCNs) and LSTM to capture spatial terms and temporal attributes and predict the future $PM_{2.5}$ concentrations. Wang et al. [23] proposed a $PM_{2.5}$-GNN model, which incorporated the domain knowledge into graph-structure data to model long-term spatiotemporal dependencies, for $PM_{2.5}$ concentrations prediction. Since multiple features were considered, this model could achieve excellent prediction performance, especially for long-term predictions.

Appendix C. Related work about dynamic graph models

Recently, to better model contextual information, dynamic graph models have been employed by some researchers. Zhou et al. [13] modeled a dynamic directed graph based on the wind field among the air quality stations. They then used the GCNs to

capture the dynamic relationships among the stations and applied a temporal convolutional network (TCN) to predict the $PM_{2.5}$ concentrations. Diao et al. [24] employed a dynamic Laplacian matrix estimator to model the dynamic graph, which can better model the spatial dependencies. Based on the dynamic estimator, they proposed a dynamic spatiotemporal graph convolutional neural network for traffic forecasting and outperformed the baselines. Peng et al. [25] employed reinforcement learning to generate dynamic graphs and combined the graphs with the LSTM model for long-term traffic flow prediction. They further proved that dynamic graphs reduced the effects of data defects with extensive experiments.

Appendix D. Related work about adaptive graph learning models

To overcome the limitations of prior information, Wu et al. [26] developed an adaptive dependency matrix through node embedding to capture the hidden spatial dependency in the data. And the model Multivariate Time Series Forecasting with Graph Neural Networks (MTGNN) [27] also used this method to extract the uni-directed relations among variables. However, this method of changing the adjacency matrix with the time of the event will bring a lot of interference information to the training of the model, thereby affecting the accuracy of the prediction. Therefore, Graph WaveNet does not perform well on actual air quality prediction datasets.

Appendix E. Other information

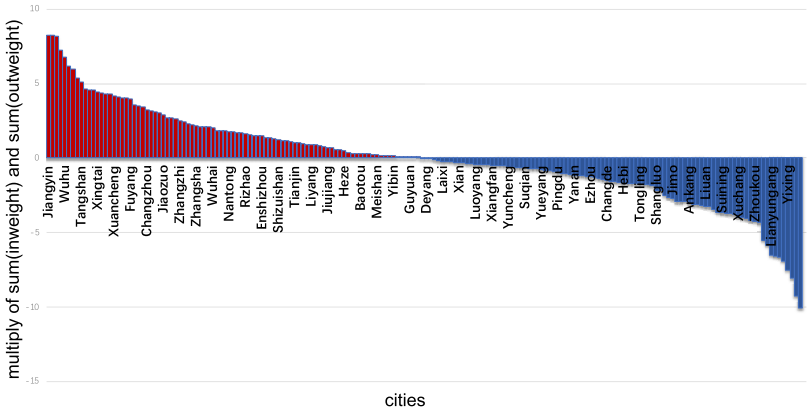


Fig. E.8. Cities sorted by the multiply of sum (in-weight) and sum (out-weight). The left red-colored cities mean they tend to inflect their neighbors, and the right blue ones are just the opposite.

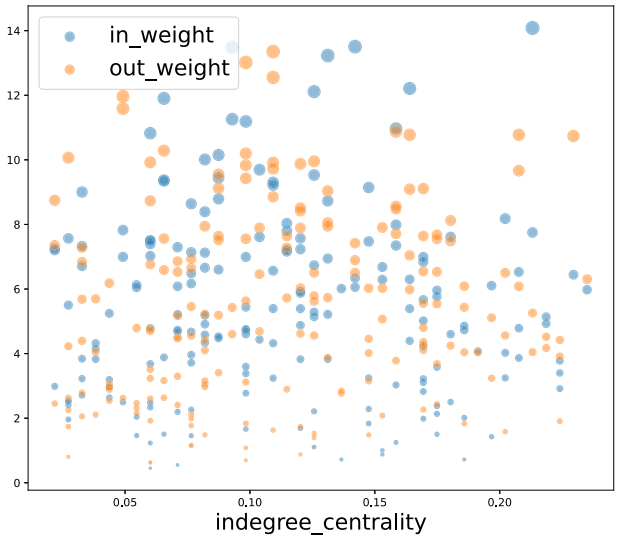


Fig. E.9. Relationships between degree centrality and connection weights.

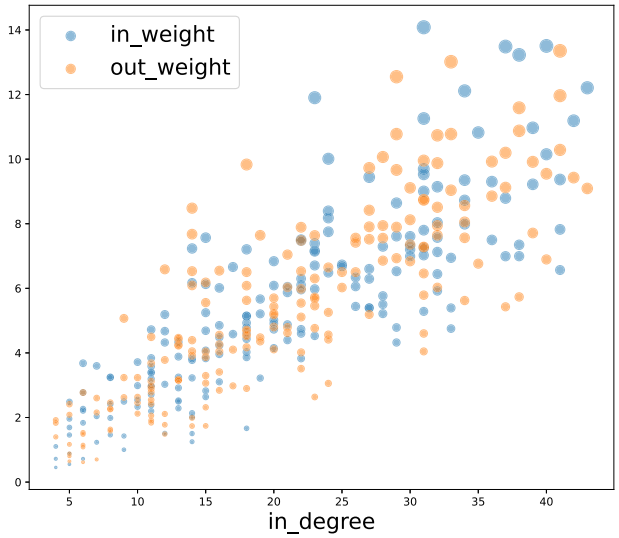


Fig. E.10. Relationships between degree and connection weights.

$$C_D\left(N_i\right) = \sum_{J=1}^{g} x_{ij}\,(i \neq j) \tag{E.1}$$

$$C\prime_D\left(N_i\right) = \frac{C_D\left(N_i\right)}{g-1} \tag{E.2}$$

Table E.3
Prediction accuracy compared with Graph WaveNet(w/o weather).

Dataset	Methods / Metric		Graph-WaveNet (w/o weather)	DGN-AEA (w/o weather)
Dataset 1	RMSE	3	24.22±0.03	**13.03±0.06**
		6	29.55±0.05	**17.08±0.07**
		12	35.14±0.23	**20.78±0.10**
		24	37.74±0.08	**24.41±0.08**
	MAE	3	14.74±0.01	**10.14±0.05**
		6	18.19±0.02	**13.73±0.06**
		12	22.11±0.15	**16.90±0.11**
		24	24.13±0.01	**19.84±0.09**
Dataset 2	RMSE	3	34.92±0.01	**19.62±0.04**
		6	42.58±0.02	**26.17±0.03**
		12	52.93±0.04	**32.87±0.07**
		24	60.46±0.05	**38.89±0.08**
	MAE	3	23.23±0.01	**15.31±0.04**
		6	29.29±0.02	**21.12±0.03**
		12	37.33±0.03	**26.99±0.08**
		24	43.32±0.05	**31.94±0.11**
Dataset 3	RMSE	3	47.11±0.03	**27.16±0.05**
		6	58.49±0.04	**36.12±0.14**
		12	69.95±0.08	**44.91±0.13**
		24	75.30±0.08	**49.71±0.11**
	MAE	3	31.04±0.01	**21.15±0.05**
		6	39.93±0.03	**29.11±0.14**
		12	49.99±0.04	**36.93±0.16**
		24	54.09±0.06	**41.44±0.13**

References

[1] P. Strøm-Tejsen, D. Zukowska, P. Wargocki, D.P. Wyon, The effects of bedroom air quality on sleep and next-day performance, Indoor Air 26 (5) (2016) 679–686.

[2] T. Vandyck, K. Keramidas, A. Kitous, J.V. Spadaro, R. Van Dingenen, M. Holland, B. Saveyn, Air quality co-benefits for human health and agriculture counterbalance costs to meet Paris agreement pledges, Nat. Commun. 9 (1) (2018) 1–11.

[3] H. Zhang, S. Wang, J. Hao, X. Wang, S. Wang, F. Chai, M. Li, Air pollution and control action in Beijing, J. Clean. Prod. 112 (2016) 1519–1527.

[4] L. Li, Y. Lei, S. Wu, Z. Huang, J. Luo, Y. Wang, J. Chen, D. Yan, Evaluation of future energy consumption on $PM_{2.5}$ emissions and public health economic loss in Beijing, J. Clean. Prod. 187 (2018) 1115–1128.

[5] X. Lu, C. Lin, W. Li, Y. Chen, Y. Huang, J.C. Fung, A.K. Lau, Analysis of the adverse health effects of pm2. 5 from 2001 to 2017 in China and the role of urbanization in aggravating the health burden, Sci. Total Environ. 652 (2019) 683–695.

[6] B. Leclercq, J. Kluza, S. Antherieu, J. Sotty, L. Alleman, E. Perdrix, A. Loyens, P. Coddeville, J.-M.L. Guidice, P. Marchetti, et al., Air pollution-derived $PM_{2.5}$ impairs mitochondrial function in healthy and chronic obstructive pulmonary diseased human bronchial epithelial cells, Environ. Pollut. 243 (2018) 1434–1449.

[7] X. Sun, C. Liu, Z. Wang, F. Yang, H. Liang, M. Miao, W. Yuan, H. Kan, Prenatal exposure to residential $PM_{2.5}$ and anogenital distance in infants at birth: a birth cohort study from Shanghai, China, Environ. Pollut. 264 (2020) 114684.

[8] G. Li, L. Li, D. Liu, J. Qin, H. Zhu, Effect of $PM_{2.5}$ pollution on perinatal mortality in China, Sci. Rep. 11 (1) (2021) 1–12.

[9] X. Yi, J. Zhang, Z. Wang, T. Li, Y. Zheng, Deep distributed fusion network for air quality prediction, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 965–973.

[10] B.N. Murphy, C.G. Nolte, F. Sidi, J.O. Bash, K.W. Appel, C. Jang, D. Kang, J. Kelly, R. Mathur, S. Napelenok, et al., The detailed emissions scaling, isolation, and diagnostic (desid) module in the community multiscale air quality (cmaq) modeling system version 5.3, Geosci. Model Dev. Discuss. 2020 (2020) 1–28.

[11] M. Ibrahim, Air quality analyses for photochemical smog associated with atmospheric aerosol particles and ozone precursors using cmaq and camx modeling systems, Int. J. Sci. Res. Sci. Technol. (2019) 224–235.

[12] J. Hong, F. Mao, Q. Min, Z. Pan, W. Wang, T. Zhang, W. Gong, Improved $PM_{2.5}$ predictions of wrf-chem via the integration of himawari-8 satellite data and ground observations, Environ. Pollut. 263 (2020) 114451.

[13] H. Zhou, F. Zhang, Z. Du, R. Liu, Forecasting $PM_{2.5}$ using hybrid graph convolution-based model considering dynamic wind-field to offer the benefit of spatial interpretability, Environ. Pollut. 273 (2021) 116473.

[14] V. Naveen, N. Anu, Time series analysis to forecast air quality indices in thiruvananthapuram district, kerala, India, Int. J. Eng. Res. Appl. 7 (6) (2017) 66–84.

[15] B.-C. Liu, A. Binaykia, P.-C. Chang, M.K. Tiwari, C.-C. Tsao, Urban air quality forecasting based on multi-dimensional collaborative support vector regression (svr): a case study of Beijing-Tianjin-Shijiazhuang, PLoS ONE 12 (7) (2017) e0179763.

[16] B. Pan, Application of Xgboost Algorithm in Hourly Pm2. 5 Concentration Prediction, IOP Conference Series: Earth and Environmental Science, vol. 113, IOP Publishing, 2018, p. 012127.

[17] R. Yu, Y. Yang, L. Yang, G. Han, O.A. Move, Raq–a random forest approach for predicting air quality in urban sensing systems, Sensors 16 (1) (2016) 86.

[18] C. Wen, S. Liu, X. Yao, L. Peng, X. Li, Y. Hu, T. Chi, A novel spatiotemporal convolutional long short-term neural network for air pollution prediction, Sci. Total Environ. 654 (2019) 1091–1099.

[19] Q. Zhang, S. Wu, X. Wang, B. Sun, H. Liu, A $PM_{2.5}$ concentration prediction model based on multi-task deep learning for intensive air quality monitoring stations, J. Clean. Prod. 275 (2020) 122722.

[20] Y. Lin, N. Mago, Y. Gao, Y. Li, Y.-Y. Chiang, C. Shahabi, J.L. Ambite, Exploiting spatiotemporal patterns for accurate air quality forecasting using deep learning, in: Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2018, pp. 359–368.

[21] J. Han, H. Liu, H. Zhu, H. Xiong, D. Dou, Joint air quality and weather prediction based on multi-adversarial spatiotemporal networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 4081–4089.

[22] C. Wang, Y. Zhu, T. Zang, H. Liu, J. Yu, Modeling inter-station relationships with attentive temporal graph convolutional network for air quality prediction, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021, pp. 616–634.

[23] S. Wang, Y. Li, J. Zhang, Q. Meng, L. Meng, F. Gao, $PM_{2.5}$-gnn: a domain knowledge enhanced graph neural network for $PM_{2.5}$ forecasting, in: Proceedings of the 28th International Conference on Advances in Geographic Information Systems, 2020, pp. 163–166.

[24] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, S. He, Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 890–897.

[25] H. Peng, B. Du, M. Liu, M. Liu, S. Ji, S. Wang, X. Zhang, L. He, Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning, Inf. Sci. 578 (2021) 401–416.

[26] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial-temporal graph modeling, arXiv preprint, arXiv:1906.00121, 2019.

[27] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang, Connecting the dots: multivariate time series forecasting with graph neural networks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 753–763.

[28] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.

[29] J. Leskovec, J. Kleinberg, C. Faloutsos, Graphs over time: densification laws, shrinking diameters and possible explanations, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, 2005, pp. 177–187.

[30] J. You, R. Ying, X. Ren, W. Hamilton, J. Leskovec, Graphrnn: generating realistic graphs with deep auto-regressive models, in: International Conference on Machine Learning, PMLR, 2018, pp. 5708–5717.

[31] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, Adv. Neural Inf. Process. Syst. 29 (2016).

[32] Z. Tong, Y. Liang, C. Sun, D.S. Rosenblum, A. Lim, Directed graph convolutional network, arXiv preprint, arXiv:2004.13970, 2020.

[33] Y. Ma, J. Hao, Y. Yang, H. Li, J. Jin, G. Chen, Spectral-based graph convolutional network for directed graphs, arXiv preprint, arXiv:1907.08990, 2019.

[34] J. Skarding, B. Gabrys, K. Musial, Foundations and modeling of dynamic networks using dynamic graph neural networks: a survey, IEEE Access 9 (2021) 79143–79168.

[35] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Neural message passing for quantum chemistry, in: International Conference on Machine Learning, PMLR, 2017, pp. 1263–1272.

[36] Z. Zhou, G. Lin, K. Yang, L. BAI, Y. Wang, et al., Greto: remedying dynamic graph topology-task discordance via target homophily, in: The Eleventh International Conference on Learning Representations.

[37] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: data-driven traffic forecasting, arXiv preprint, arXiv:1707.01926, 2017.

[38] S. Du, T. Li, Y. Yang, S.-J. Horng, Deep air quality forecasting using hybrid deep learning framework, IEEE Trans. Knowl. Data Eng. 33 (6) (2019) 2412–2424.

[39] Y. Qi, Q. Li, H. Karimian, D. Liu, A hybrid model for spatiotemporal forecasting of $PM_{2.5}$ based on graph convolutional neural network and long short-term memory, Sci. Total Environ. 664 (2019) 1–10.

[40] J.K. Rekhi, P. Nagrath, R. Jain, et al., Forecasting air quality of Delhi using arima model, in: Advances in Data Sciences, Security and Applications, Springer, 2020, pp. 315–325.

[41] M. Dun, Z. Xu, Y. Chen, L. Wu, Short-term air quality prediction based on fractional grey linear regression and support vector machine, Math. Probl. Eng. 2020 (2020).

[42] B. Liu, Y. Jin, C. Li, Analysis and prediction of air quality in Nanjing from autumn 2018 to summer 2019 using pcr–svr–arma combined model, Sci. Rep. 11 (1) (2021) 1–14.

[43] J. Ma, Z. Li, J.C. Cheng, Y. Ding, C. Lin, Z. Xu, Air quality prediction at new stations using spatially transferred bi-directional long short-term memory network, Sci. Total Environ. 705 (2020) 135771.