A SPATIOTEMPORAL GRAPH NEURAL NETWORK WITH MULTI GRANULARITY FOR AIR QUALITY PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Air quality prediction is a complex system engineering. How to fully consider the impact of meteorological, spatial and temporal factors on air quality is the core problem. To address this central conundrum, in an elaborate encoder-decoder architecture, we propose a new air quality prediction method based on multi-granularity spatiotemporal graph network. At the encoder, firstly, we use multi granularity graph and the well-known HYSPLIT model to build spatial relationship and dynamic edge relationship between nodes, respectively, while meteorological, temporal and topographic characteristics are used to build node features and LSTM (Long Short Term Memory) is used to learn the time-series relationship of pollutant concentration. At the decoder, secondly, we use the attention mechanism LSTM for decoding and forecasting of pollutant concentration. The proposed model is capable of tracking different influences on prediction resulting from the changes of air quality. On a project-based dataset, we validate the effectiveness of the proposed model and examine its abilities of capturing both fine-grained and long-term influences in pollutant process. We also compare the proposed model with the state-of-the-art air quality forecasting methods on the dataset of Yangtze River Delta city group, the experimental results show the appealing performance of our model over competitive baselines.

1 INTRODUCTION

Air quality which is closely related to human public health, has been a common research hotspot focused by scholars all over the world. At present, a large number of air quality monitoring stations (stations for short) have been built in major cities to monitor the concentration of air pollutants (PM2.5, PM10, O_3, etc.) and meteorological parameters (temperature, pressure, wind speed, wind direction, humidity, etc.). However, these stations only can monitor real-time air quality, and fail to provide air quality prediction and auxiliary support for urban intelligent decision-making or activity planning. How to construct an air quality prediction model using a large amount of historical monitoring data has become a hot research topic in the field of environmental engineering. Unfortunately, air quality prediction is an extremely complex system engineering. On the one hand, air quality is related to pollutant emission, which is a type of time sequence and has periodicity; On the other hand, there exist physical and chemical changes of pollutants in the air, such as diffusion and deposition, which are greatly affected by meteorological and geographical locations; Finally, air quality also has certain probability, such as unexpected pollution leakage events will lead to a sharp decline in air quality.

The commonly used types of air quality prediction include mechanism model and machine learning methods. The mechanism model method, also known as numerical model, uses atmospheric physical and chemical reactions to model the emission and diffusion process of air pollutants, and then carries out air quality prediction. For example, Gaussian diffusion models of AERMOD and ADMS, Lagrange models of CALPUFF and HYSPLIT can be applied to small-scale and medium air quality prediction Jittra et al. (2015); and the third generation air quality models Zhang (2017) such as CMAQ, CAMX, WRF-CHEM, NAQPMS, and so on can be applied to predict large-scale air quality. The above mechanism model can be applied to the prediction of pollutant concentration, but there are some problems such as low accuracy, harsh model input and limited application environment. For example, AERMOD is an empirical model which is mainly applicable to small-scale

air diffusion simulation and pollutant prediction. And the third generation air quality model needs comprehensive and accurate source list and meteorological field data as input to predict, and its application is limited.

With the development of deep learning, air quality prediction methods based on machine learning have attracted more attention Yi et al. (2018). The air quality prediction method based on machine learning takes advantage of a large number of historical observation data for training and testing, finding out the change law of pollutant concentration, and then predicts the air quality. Machine learning is a novel air quality prediction method based on data-driven strategy, which can effectively capture the regularity and time periodicity of pollutant concentration changes and has the characteristics of simplicity, flexibility and rapid deployment. However, these methods are unable to timely capture the regional impact and sudden change caused by the pollutant diffusion in the air. On the other hand, the traditional mechanism model based on the principle of atmospheric diffusion can better simulate the impact of surrounding areas on the prediction points.

Motivated by the above considerations, in this paper, we use the LSTM (Long Short Term Memory) model to learn the timing and periodicity of pollutant concentration, and use the GNN (Graph Neural Network) model to learn the spatial relationship of pollutants among stations, and thus building a multi granularity spatiotemporal graph neural network model, called MGST_GNN. In order to better capture the mutation of pollutant concentration and its impact on air diffusion and deposition, the mechanism model HYSLPLIT is used to dynamically construct the adjacency matrix and edge relation of the spatiotemporal graph.

2 RELATED WORK

For air quality prediction, the existing methods are mainly divided into two categories: mechanism model and machine learning method. The mechanism model method uses physical and chemical models to simulate the formation, diffusion, and transfer of various pollutants in the air to predict the concentration of air pollutants Jittra et al. (2015); Arystanbekova (2004); Stein et al. (2015); Wang et al. (2012). However, most of mechanism model methods require many empirical parameters and assumptions, which are prone to be reliable for a specific environment but not for all urban environments Zou et al. (2021). Machine learning methods use a data-driven process to capture the dependencies in spatial and temporal dimensions, modeling the correlations between multi-source complex factors and air quality, which include linear statistical models Moisan et al. (2018), fitting optimization techniques Niu et al. (2017), and deep learning methods Ma et al. (2020).

Since deep learning has a powerful function to automatically extract nonlinear features, recent literatures about predicting air quality often rely on deep learning models Yi et al. (2018); Zou et al. (2021); Ma et al. (2020); Wang et al. (2019). There exist spatial location relationships among different stations, so the station can be regarded as a pixel in the image. Therefore, based on a large number of stations, a station image can be constructed and processed by CNN (Convolutional Neural Network). In addition, the pollutant concentration value of the station is a typical sequence, so the LSTM network can be constructed for sequence analysis. For that reason, Zhang et al. (2016) proposed a deep learning air quality prediction method combining CNN and LSTM, which achieved good results and made scholars see the dawn of the application of deep learning in air quality prediction. Subsequently, Du et al. (2019) used one-dimensional CNN to capture local time trend and bidirectional LSTM to extract long-term time series features, and then to construct a hybrid neural network for air quality prediction. Liang et al. (2018) proposed a GeoMAN based on LSTM and encoder-decoder architecture for air quality prediction, and took advantage of the attention mechanism to capture the spatial impact relationship among different stations. Xu et al. (2018) introduced Echo State Networks to predict daily air quality. Yi et al. (2018) proposed the DeepAir model to construct a sub-network with multi-source data, and used the fusion network to integrate the results of different sub-networks to obtain the final predicted value. Considering the sparsity of stations spatial distribution, Chen et al. (2019) used PANDA, a multi-task learning framework, to perform multi-factor estimation of air quality, thus assisting in the prediction of pollutant concentration. The above methods captured temporal correlation well by LSTM, but their capture of spatial relationship was obviously insufficient. Although CNN can be used to establish spatial relationships, it is a static spatial relationship, and the distance among stations is fixed. Due to the influence of weather and terrain, the spatial relationship among stations is not a simple static distance relationship, but a

Table 1: quarter feature representing method

	Spring	Summer	Fall	Winter
spring season	1	0	0	0
summertime	0	1	0	0
autumn	0	0	1	0
wintertime	0	0	0	1

dynamic. In addition, the stations in the city are unevenly distributed and sparse, so interpolation is required in the construction of CNN, resulting in a large number of virtual stations, which will affect the prediction results.

In contrast, graph-based models naturally sidestep the above issue since they shape the concentration values into graph nodes and keep their original distributions in graph structures. Because graph has the ability to construct non Euclidean entity distribution, it can capture spatial relationships well. Thus, in order to compensate for the lack of spatial relationship learning in the above methods, the methods based on GNN (Graph Neural Network) are applied to air quality prediction. Qi et al. (2019) used GNN to learn the spatial relationship among stations, and LSTM to learn the time correlation of stations, so as to build a comprehensive prediction model GC-LSTM. Lin et al. (2018) used diffusion convolution operation to replace matrix multiplication in GRU (Gate Recurrent Unit) for sequence modeling, and combined with graph convolution operation to build GC-DCRNN for air quality prediction. Xu et al. (2020) proposed ST-MFGCN for air quality prediction. Its main innovation is to obtain the spatiotemporal variation law of vehicle emissions by building a graph structure traffic network, and then predict traffic pollution emissions. Xu et al. (2021) proposed HighAir, i.e., a hierarchical graph neural network-based air quality forecasting method, which adopted an encoder-decoder architecture and considered complex air quality influencing factors, e.g., weather and land usage.

The above methods based on GNN use graph structure to effectively construct the spatial relationship among stations, but they fail to fully construct the edges in the graph structure. For example, literature Xu et al. (2021) simply used distance and wind direction similarity to construct edge weights. In order to better build the edge relationship, Wang et al. (2020) identified a set of critical domain knowledge for PM2.5 forecasting and developed a novel graph based model, PM2.5-GNN,they used domain knowledge (wind speed and direction, distance, advection coefficient) to construct edge weights. Although they make use of domain knowledge, they simply list some impact factors, which is not enough. The pollution impact relationship among stations is comprehensively determined by meteorological conditions (wind speed, wind direction, pressure, humidity, planetary boundary layer height, etc.) and landform. It is a complex process and needs to be analyzed by using professional models.

3 PRELIMINARY

3.1 RELATED CONCEPTS AND PROBLEM DEFINITION

Definition 1 (Region and monitoring stations): we set $R = \{r_a, 1 \leq a \leq N\}$ as a set of N regions, $L = \{l_a, 1 \leq a \leq N\}$ as a location set of N regions, $S_a = \{s_{a,i}, 1 \leq i \leq |S_a|\}$ as a set of air quality monitoring stations in region r_a, $\mathcal{L}_a = \{l_{a,i}, 1 \leq i \leq |S_a|\}$ as a location set of monitoring stations in region r_a, where $l_{a,i}$ is composed of longitude and latitude of a station, and l_a is the mean value of $l_{a,i}$ in $\mathcal{L}_a$. Note: According to different application scenarios, the region referred to in this paper can be a national region, a city region or an administrative region, etc.

Definition 2 (Time Feature): Time Feature (TF) includes five features: quarter, month, week, hour and holiday. We represent the quarter by using one-hot encoding adopted for 4-bit binary representation, as shown in Table 1. Similarly, the month is represented with 12-bit binary by one-hot encoding; week is represented with 7-bit binary by one-hot encoding; Hour is represented with 24-bit binary code by one-hot coding mode; Holiday is represented with 2-bit binary by one-hot coding (0 means non-holidays, 1 means holiday).

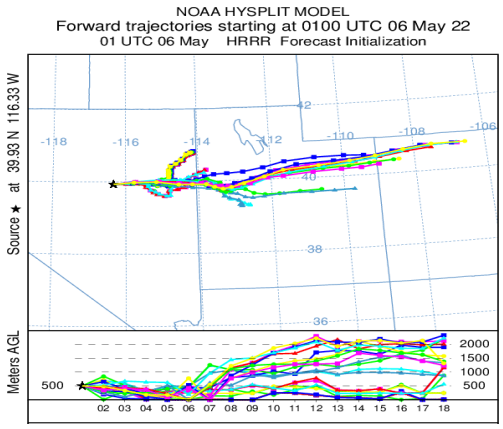


Figure 1: Trajectory analysis chart

Definition 3 (Geomorphic Feature): Geomorphic feature (GF) contains the topography and land usage information of a monitoring station, which consider the altitude and five land usage categories: residential area, park, mountain, water (river or pool), and industry. Where, the altitude is divided into four categories: very high (more than 1300m), high (1000-1300m), medium (500-1000m) and low (less than 500m), and the corresponding categories are represented by the numbers 1, 2, 3 and 4; The land usage type is determined by the number of major land use types within 10km around the station. For example, the altitude of $s_{a,i}$ is 800 meters, and there is one residential area, two park, one mountain, three pool, and two industrial facilities within the perception radius 10 kilometers of $s_{a,i}$. Thus, the GF vector $gf_{a,i}$ can be represented as $[3, 1, 2, 1, 3, 2]$.

Definition 4 (Weather Data): The weather data of region r_a and monitoring stations $s_{a,i}$ at time slot t is represented as a vector wd_a^t and $wd_{a,i}^t$, including temperature, humidity, rainfall, wind speed, wind direction, and air pressure.

Definition 5 (Pollutant concentration data): The pollutant concentration data includes the concentrations of six major pollutants such as PM2.5, PM10, SO_2, CO_2, CO and O_3. Among them, PM2.5 and O_3 are the most concerned at present. Therefore, the later experiments focus on the concentration prediction of these two pollutants.

Definition 6 (HYSPLIT): HYSPLIT Stein et al. (2015); Warner (2018) is a complete system for computing simple air parcel trajectories, as well as complex transport, dispersion, chemical transformation, and deposition simulations. HYSPLIT continues to be one of the most extensively used atmospheric transport and dispersion models in the atmospheric sciences community. A common application is a back trajectory analysis to determine the origin of air masses and establish source-receptor relationships. HYSPLIT has also been used in a variety of simulations describing the atmospheric transport, dispersion, and deposition of pollutants and hazardous materials.

In this paper, HYSPLIT is used to establish the source-receptor relationship among nodes by trajectory analysis. When HYSPLIT is used for trajectory analysis, it only needs to input the meteorological data of the simulation area and the coordinate information of the initial point of the simulation. The meteorological data can be downloaded from the official website of Air Resources Laboratory (ARL) [1] for free. Fig. 1 shows an example of trajectory analysis using HYSPLIT. The example takes Beijing Center $(116°20', 39°56')$ as the starting point and 10:00 on May 6, 2022 as the starting time to predict the air mass trajectory in the next 48 hours. It can be seen from Fig. 1 that the location and time of each track can be obtained by HYSPLIT.

3.2 Problem Statement

Air quality prediction task is described as follows: Given region locations L, station locations $\mathcal{L}$, GF data $gf_{a,i}$, τ_{in} hours of pollutant concentration data con, and $\tau_{in} + \tau_{out}$ hours of weather data wd, the air quality prediction task aims to forecast the pollutant concentrations of stations for the

[1] https://nomads.ncep.noaa.gov/pub/data/nccf/com/hysplit

next τ_{out} hours, where τ_{in} denotes the length of historical time window and τ_{out} denotes forecasting horizon.

4 METHODOLOGY

4.1 POLLUTANT CONCENTRATION PREDICTION MODEL FRAMEWORK

Existing air quality prediction methods based on GNN Du et al. (2019); Lin et al. (2018) mostly simulate the spatial relations among stations by constructing a flat static graph. However, due to the distance among stations and terrain, it is difficult to transfer information among stations in different regions through flat graph. In addition, the spatial relationship among stations is greatly affected by the dynamic wind direction, wind speed and other meteorology, so it is difficult to construct its edge relationship by means of static graph. Finally, pollutant concentrations at different time in history have different impacts on current air quality.

In order to overcome the shortcomings of flat static graphs, we build an encoder-decoder architecture based on dynamic multi-granularity spatiotemporal graph by referring to literature Wang et al. (2020), as shown in Fig. 2. In the encoding stage, multi-granularity graph network is used to learn the spatial relationship among stations and LSTM network is used to learn its temporal relationship. In the decoding stage, auxiliary data and attention mechanism are used to enhance LSTM prediction learning and decode the future pollutant concentration value. This framework fully considers three key factors affecting air quality, namely, meteorology, space and time. Among them, the multi-granularity spatiotemporal graph neural network (as shown in Fig. 3) is the core component.

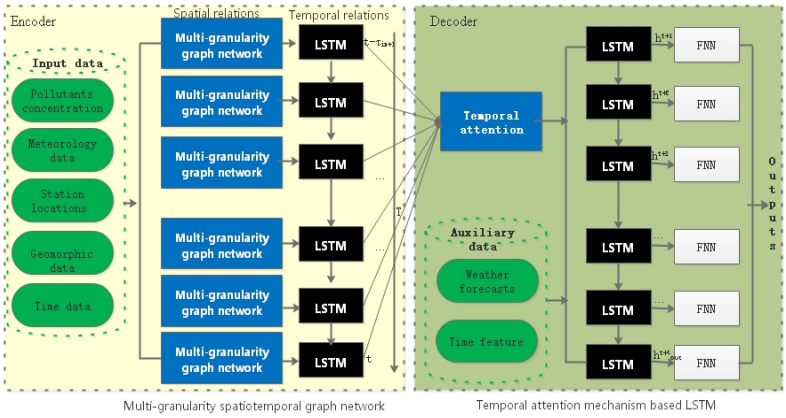


Figure 2: Framework diagram of pollutant concentration prediction model

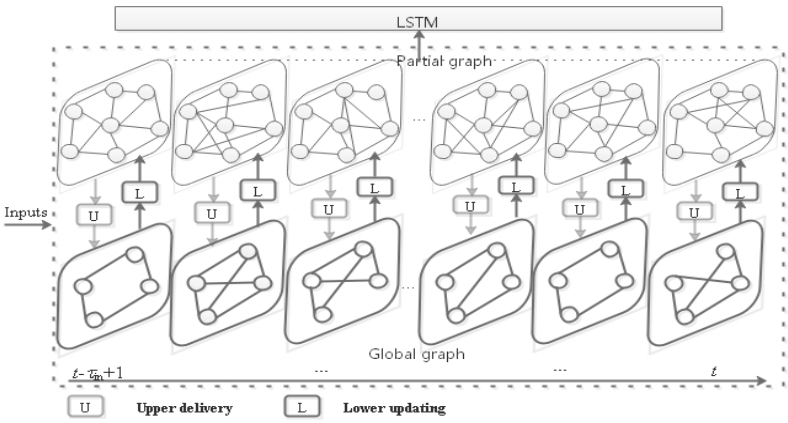


Figure 3: Multi-granularity spatiotemporal graph neural network

4.2 SPATIOTEMPORAL GRAPH CONSTRUCTION

We propose a multi-granularity spatiotemporal graph model (MGST_GNN) by referring to reference Xu et al. (2021). Assume that V is the set of nodes, E is the set of connected edges among nodes, and u is the global attribute of a station-level partial subgraph. The MGST_GNN can be used to predict the air quality of all stations in a city or all stations in the whole country. MGST_GNN can be divided into global and partial granularity graph structures. Global graph and partial subgraph interact through message passing mechanism. Partial subgraph's information (including current and historical pollutant concentration information) is upper delivered to the global graph; the global graph's information (including global pollutant concentration information) is transmitted downward to the partial subgraph. In each time period, MGST_GNN will calculate the attribute information of each global node and use it to update the corresponding partial node attribute. For each region of the partial subgraph, we calculate the overall air quality information representation of the regions at each time slot, forming a sequence sent to the LSTM to get the representations of current and historical pollutant concentration information.

Taking the future air quality prediction of all air monitoring stations in a city as an example, al-l stations in the city are clustered according to their longitude and latitude information to obtain N categories, that is, N regions are clustered. Therefore, each region can be regarded as a node in the global graph, while each station in the region can be regarded as a node in the partial subgraph. A graph composed of region-level nodes is called a global graph, while a graph composed of station-level nodes is called a partial subgraph. From this, a multi-granularity graph network can be constructed, as shown in Fig. 4, where the coarse-grained graph is a region-level global graph and the fine-grained graph is a station-level partial subgraph.

4.2.1 ATTRIBUTE DEFINITION OF GRAPH NODE

To the node in the global graph, its attribute is the pollutant concentration value of the node region. Correspondingly, to the node in partial subgraph, its attribute is the pollutant concentration value of the node station, and its geomorphic feature, plus the time feature of quarter, month, week, hour and holiday.

4.2.2 EDGE AND ATTRIBUTE DEFINITIONS OF GRAPHS

Wind speed, wind direction, rainfall and other meteorological data have a decisive influence on the horizontal transmission of pollutants. In order to make use of this domain knowledge, we adopt the air quality model HYSPLIT to learn the relationship among nodes and build dynamic connection edges. Taking the edge calculation of global graph as an example, the city is divided into grids according to the clustered regions, so that each region falls into a unique grid. For each region (node v_a), HYSPLIT is used to calculate all trajectories starting from v_a and stepping in hours within the next time t. With HYSPLIT, we can track the time when each trajectories crosses the grid where the node is located, and record the number of pass trajectories and crossing time of the grid from other nodes except v_a, and dynamically construct the global connection edge and its attribute vector. Repeating these steps until each station is analyzed as a starting point. For example, taking v_1 as the starting point, using HYSPLIT to calculate that there are 5 trajectories passing through v_2 in the following 48 hours, and the time interval is 1 hour, 3 hours, 4 hours, 8 hours and 48 hours in the future, then the edge relationship between node v_1 and v_2 is formed, and its attribute vector is: $e_{12} = [1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]$. From e_{12}, we can get the specific impact time of node v_1 to v_2. It is a dynamic, specific and accurate edge attribute, which is calculated by the mechanism model according to the meteorological conditions and geographical location. Station-level partial subgraph edge and attribute calculation are similar.

4.3 MODELING OF SPATIAL DEPENDENCIES

MGST_GNN can model spatial dependencies with different granularities by message passing mechanism, which mainly includes message aggregation and description updating. Message aggregation is shown as follows:

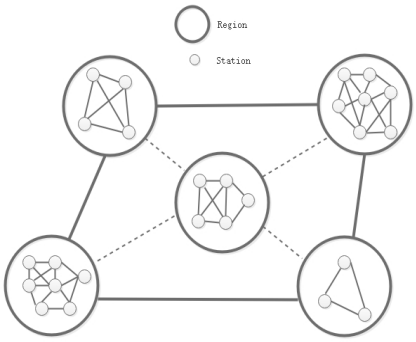


Figure 4: Multi-granularity graph structure

$$R_a = \{x_s, x_a, e_{a,s}\}_{s \in N_a} \tag{1}$$

$$r_a \leftarrow \Psi(R_a) \tag{2}$$

where R_a denotes the set of all the messages passed to node v_a; x_a is the attribute of node v_a; x_s is the attribute of a neighbor node; $e_{a,s}$ is the edge attribute; N_a denotes the neighbor node net of node v_a (dynamically determined by HYSPLIT trajectory); r_a is the aggregation vector of node v_a; $\Psi(\cdot)$ denotes aggregate function.

Description updating is shown as follows

$$\begin{cases} x_a^{'} \leftarrow \phi_1(r_a, x_a), & \text{in global graph} \\ x_{a,i} \leftarrow \phi_2(r_{a,i}, x_{a,i}, u_a), & \text{in partial graph} \end{cases} \tag{3}$$

where ϕ_1 and ϕ_2 denote update functions; $r_{a,i}$ is the aggregation vector of node $v_{a,i}$ in the a-th partial subgraph; $x_{a,i}$ is the attribute of node $v_{a,i}$; u_a is the global attribute of a-th partial subgraph; ϕ_1 and ϕ_2 can be implemented by using different FNNs.

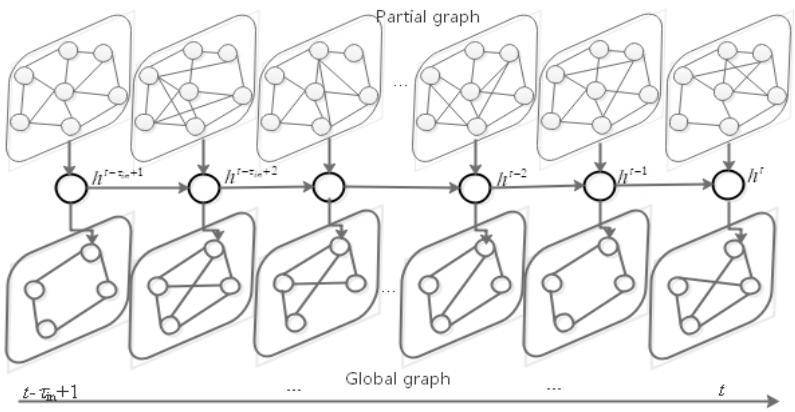


Figure 5: Message aggregation diagram

The specific message aggregation method is from the partial subgraph to the global graph, as shown in Fig. 5. The information transmitted from the partial subgraph includes the current and historical pollutant concentration values, and the average method is adopted to aggregate the information transmitted from the partial subgraph:

$$con_a^t = \text{mean}(con_{a,i}^t), \quad 1 \leq i \leq \mid S_a \mid \tag{4}$$

where con_a^t denotes the aggregate value at time t of node v_a in the global graph; $con_{a,i}^t$ denotes the pollutant concentration value at time t of node $v_{a,i}$ in the a-th partial subgraph; S_a is the number of nodes in the a-th partial subgraph.

For nodes in the global graph, message aggregation is used to calculate the global representation of pollutant concentration in each time slot, thus forming a series of sequences $\{con_a^{t-\tau_{in}+1}, con_a^{t-\tau_{in}+2}, \ldots, con_a^t \mid 1 \leq a \leq N\}$. These sequences are then fed into the global LSTM to learn the current and historical representations. The notation h in Fig. 5 represents the hidden layer state of each time slot in LSTM, and is taken as the initial attribute of nodes in the global graph, so the global graph includes current and historical information.

The specific updating method is to update the partial subgraph by using the global graph, as shown in Fig. 6. The information transmitted from the global graph includes the historical pollutant concentration values of all regions. At each time slot, the output of the global graph is fed into the FNN to obtain a downward update vector, which is used to update the global attribute u of the partial subgraph. The u includes meteorological parameters and downward update vector, which are used in the partial subgraph by message passing. Therefore, the nodes in the station-level partial subgraph can make use of the historical information of neighbor nodes in the region-level global graph.

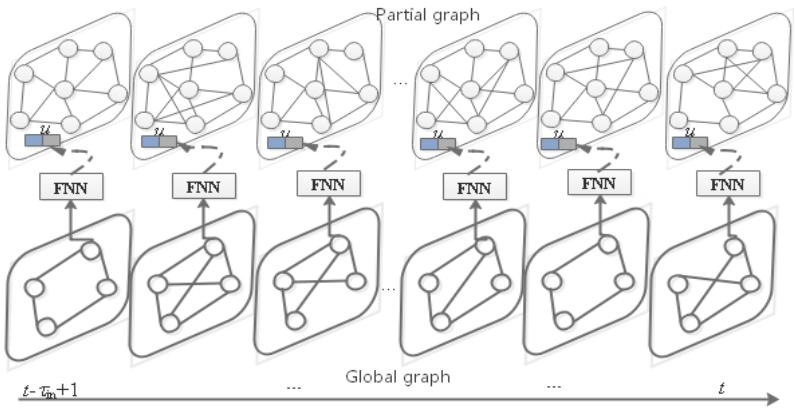


Figure 6: Description updating chart

4.4 Modeling of temporal dependencies

In order to capture the temporal dependence, we adopt an encoder-decoder architecture, as shown in Fig. 1. In each time slot, node attributes in the partial subgraph form a sequence $X_{a,i} = \{x_{a,i}^{t-\tau_{in}+1}, x_{a,i}^{t-\tau_{in}+2}, \ldots, x_{a,i}^t \mid 1 \leq a \leq N, 1 \leq i \leq |S_a|\}$. That is, each time slot in the historical has a corresponding node attribute. The LSTM in the encoder uses $X_{a,i}$ as the input and the final state of the LSTM is used as the input of the decoder. The input of the LSTM in the decoder includes not only the output of the encoder, but also the node attribute in the partial subgraph. The output of LSTM in the decoder is used as the input of FNN, and FNN outputs the predicted pollutant concentration value in the future time slot.

In order to make better use of temporal characteristics, temporal attention mechanism is introduced in the decoding stage to learn the dynamic temporal correlation between future time and historical time. Give the hidden state $h_{t'-1}^{'}$ and cell state $c_{t'-1}^{'}$ of the LSTM in the decoder at time $t^{'}-1$, then at time $t^{'}$, the attention weight of the hidden state h_t output by the encoder is calculated as follows:

$$\phi_{t'}^t = v^T \tanh(W[h_{t'-1}^{'}; c_{t'-1}^{'}] + Uh_t + b) \tag{5}$$

$$\lambda_{t'}^t = \frac{\exp(\phi_{t'}^t)}{\sum_{t=1}^{T} \exp(\phi_{t'}^t)} \tag{6}$$

where $\lambda_{t'}^{t}$ is the attention weight; v, b, W, U are the parameters to be learned. Through the above formulas 5 and 6, the attention weight of all historical hidden states in the encoder can be calculated, and then, the hidden state h_t is weighted and summed to obtain the time context vector c

$$c_{t'} = \sum_{t=1}^{T} \lambda_{t'}^{t} h_t \tag{7}$$

The output result $\widehat{y}_{t'-1}$ of decoder at time $t^{'} - 1$, the meteorological parameter $wd_{t'}$ at time $t^{'}$, the time feature $tf_{t'}$ and the time context vector $c_{t'}$ are connected as the input for the LSTM of the decoder at time $t^{'}$, and it is used to update the hidden state $h_{t'}^{t}$:

$$h_{t'}^{t} = \text{LSTM}(h_{t'-1}^{t}, [\widehat{y}_{t'-1}; wd_{t'}; tf_{t'}; c_{t'}]). \tag{8}$$

5 Experiments

We conduct experiments on real project database (JINAN DATABASE) and public database (YANGTZE RIVER DELTA CITY GROUP DATABASE) to verify the effectiveness and advantages of the proposed method. For detailed experimental database and setup, please refer to the Appendix Section A.1 and A.2.

Data Augmentation: We use Gaussian noise and random perturbation methods to enhance the training samples by three times. The experimental results show that the MAE (mean absolute error) of PM2.5 and O_3 decreases about 1.88 and 1.52 respectively by using the data enhancement method. See Appendix A.3 for a detailed description.

Multi-Source factor: Meteorological, temporal and spatial information are the three core factors in the air quality prediction model. The experimental results show that the MAE can be reduced about 2.55, 1.43 and 1.41 respectively by making full use of meteorological, temporal and spatial information. See Appendix A.4 for a detailed description.

Model component: The innovation of this model includes three components: multi granularity graph structure, dynamic edge based on HYSPLIT and LSTM of attention mechanism. The experimental results show that MAE decreases about 1.99, 2.09 and 0.59 respectively by using the above three components. See Appendix A.5 for a detailed description.

COMPARISON: we compare the proposed model with six state-of-the-art methods. The experimental results show that this model achieves the best performance. See Appendix A.6 for a detailed description.

6 Conclusion

In this paper, the multi-influences of meteorological, spatial and temporal factors on air quality prediction are fully considered, and an encoder-decoder architecture based on multi-granularity spatiotemporal graph network is proposed to predict pollutant concentration over a long period of time. Compared with the existing models, the striking characteristic of this paper is that the meteorological, spatial terrain and time factors are considered comprehensively through the professional air quality model, while other models take the influencing factors as the splitting parameter input. That is, the model in this paper integrates the advantages of mechanism model and machine learning, and namely it is a comprehensive model. The experimental results show that the proposed model provides an impressive applicability with competitive prediction performance.

References

N Kh Arystanbekova. Application of gaussian plume models for air pollution simulation at instantaneous emissions. *Mathematics and Computers in Simulation*, 67(4-5):451–458, 2004.

Ling Chen, Yifang Ding, Dandan Lyu, Xiaoze Liu, and Hanyu Long. Deep multi-task learning based urban air quality index modelling. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(1):1–17, 2019.

Shengdong Du, Tianrui Li, Yan Yang, and Shi-Jinn Horng. Deep air quality forecasting using hybrid deep learning framework. *IEEE Transactions on Knowledge and Data Engineering*, 33(6):2412–2424, 2019.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Nattawut Jittra, Nattaporn Pinthong, and Sarawut Thepanondh. Performance evaluation of AERMOD and CALPUFF air dispersion models in industrial complex area. *Air, Soil and Water Research*, 8:ASWR–S32781, 2015.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. GeoMAN: Multi-level attention networks for geo-sensory time series prediction. In *IJCAI*, volume 2018, pp. 3428–3434, 2018.

Yijun Lin, Nikhit Mago, Yu Gao, Yaguang Li, Yao-Yi Chiang, Cyrus Shahabi, and José Luis 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*, pp. 359–368, 2018.

Jun Ma, Zheng Li, Jack CP Cheng, Yuexiong Ding, Changqing Lin, and Zherui Xu. Air quality prediction at new stations using spatially transferred bi-directional long short-term memory network. *Science of The Total Environment*, 705:135771, 2020.

Stella Moisan, Rodrigo Herrera, and Adam Clements. A dynamic multiple equation approach for forecasting PM2.5 pollution in Santiago, Chile. *International Journal of Forecasting*, 34(4): 566–581, 2018.

Mingfei Niu, Kai Gan, Shaolong Sun, and Fengying Li. Application of decomposition-ensemble learning paradigm with phase space reconstruction for day-ahead PM2.5 concentration forecasting. *Journal of environmental management*, 196:110–118, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Yanlin Qi, Qi Li, Hamed Karimian, and Di Liu. A hybrid model for spatiotemporal forecasting of PM2.5 based on graph convolutional neural network and long short-term memory. *Science of the Total Environment*, 664:1–10, 2019.

AF Stein, Roland R Draxler, Glenn D Rolph, Barbara JB Stunder, MD Cohen, and Fong Ngan. NOAA's HYSPLIT atmospheric transport and dispersion modeling system. *Bulletin of the American Meteorological Society*, 96(12):2059–2077, 2015.

K Wang, Y Zhang, Athanasios Nenes, and C Fountoukis. Implementation of dust emission and chemistry into the community multiscale air quality modeling system and initial application to an asian dust storm episode. *Atmospheric Chemistry and Physics*, 12(21):10209–10237, 2012.

Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. PM2.5-GNN: A domain knowledge enhanced graph neural network for PM2. 5 forecasting. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, pp. 163–166, 2020.

Yun Wang, Guojie Song, Lun Du, and Zhicong Lu. Real-time estimation of the urban air quality with mobile sensor system. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13 (5):11–19, 2019.

Mellissa SC Warner. Introduction to PySPLIT: A Python toolkit for NOAA ARL's HYSPLIT model. *Computing in Science & Engineering*, 20(5):47–62, 2018.

Jiahui Xu, Ling Chen, Mingqi Lv, Chaoqun Zhan, Sanjian Chen, and Jian Chang. HighAir: A hierarchical graph neural network-based air quality forecasting method. *arXiv preprint arXiv:2101.04264*, 2021.

Meiling Xu, Yuanzhe Yang, Min Han, Tie Qiu, and Hongfei Lin. Spatio-temporal interpolated echo state network for meteorological series prediction. *IEEE transactions on neural networks and learning systems*, 30(6):1621–1634, 2018.

Zhenyi Xu, Yu Kang, Yang Cao, and Zhijun Li. Spatiotemporal graph convolution multifusion network for urban vehicle emission prediction. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3342–3354, 2020.

Xiuwen Yi, Junbo Zhang, Zhaoyuan Wang, Tianrui Li, and Yu Zheng. Deep distributed fusion network for air quality prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 965–973, 2018.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. ST-UNet: A spatio-temporal U-network for graph-structured time series modeling. *arXiv preprint arXiv:1903.05631*, 2019.

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

Yang Zhang. Air quality modelling: Current status, major challenges and future prospects. *Air Quality and Climate Change*, 51(3):41–47, 2017.

Xiangyu Zou, Jinjin Zhao, Duan Zhao, Bin Sun, Yongxin He, and Stelios Fuentes. Air quality prediction based on a spatiotemporal attention mechanism. *Mobile Information Systems*, 2021, 2021.

A Appendix

A.1 Experimental datasets

A.1.1 Jinan database (JN)

Jinan database is a city-level data set collected by us in projects. Jinan is located in the middle of Shandong Province, China. Its geographical position is between $36°01' \sim 37°32'$ N and $116°11' \sim 117°44'$ E. It is distributed in a narrow and long terrain with a total area of 7998 square kilometers. There are 130 air monitoring stations in Jinan. Each station outputs the concentration values of pollutants (PM2.5, PM10, SO_2, CO_2, CO, O_3) and meteorological parameters (rainfall, surface pressure, temperature, humidity, wind speed and wind direction) every hour. We collected the historical monitoring data of 130 stations in Jinan from January 1st, 2019, to January 1st, 2022, as the training and test set.

We divide the 130 stations into 13 regions, and the global graph consists of 13 regions. Each region is used as a global node in the global graph. The stations in each region form a partial subgraph, and the stations in the region are the nodes of the partial subgraph.

A.1.2 Yangtze River Delta city group database(YRD)

The city group contains ten cities: Shanghai, Hangzhou, Suzhou, Ningbo, Shaoxing, Jiaxing, Wuxi, Zhoushan, Nantong, and Huzhou. We used Air Pollution Prediction system [2] to collect historical pollutant concentration values and meteorological parameters of corresponding stations, and the time span was from January 1th, 2019, to December 31th, 2022. Therefore, we collected 3 years of historical monitoring data as the training and test set. Each city in the Yangtze River Delta city group is a global graph node, and each station in the city is a node of the corresponding partial subgraph.

Geographic features and weather forecast data are obtained by:

[2] http://airprediction.urban-computing.com

Table 2: List of sample augmentation

Sample type \ Augmentation type	Meteorological data plus noise	Concentration value disturbance
original sample	-	-
sample 1	✓	-
sample 2	-	✓
sample 3	✓	✓

- Geographic features are collected from the map engine of AMAP [3].The perception radius is set to 1000 m.
- Weather forecast data are collected from the Air Resources Laboratory (ARL)[4]. This website can download the meteorological forecast data for the following 26 days at most, with an accuracy of 0.25×0.25 degrees, which can be updated four times a day.

We use min-max normalization to normalize the pollutant concentration and meteorological parameters into [0, 1]. In the evaluation, we rescale the predicted values back to the normal values. We use interpolation method to fill the missing values in the two datasets. Sliding windows (1 step = 1 hour) are used to generate samples.

Data Augmentation: aiming at the difficulty of collecting training samples for air quality prediction, we use the method of data augmentation to reconstruct samples. Due to the distribution difference between real weather data and forecast weather data, Gaussian noise is introduced to the meteorological data in the sample to enhance the data and improve the generalization ability of the model. In addition, due to the influence of monitoring instruments, environment and other factors, the pollutant concentration values collected may be biased. Therefore, the pollutant concentration values of samples are disturbed up and down by 1 metric to enhance the number of samples and improve the model generalization ability. Therefore, Gaussian noise is introduced into meteorological data and random perturbation is for pollutant concentration data to enhance the samples by three times. The specific augmentation methods are shown in Table 2. Finally, 26136 samples were obtained from Jinan database and Yangtze River Delta city group database respectively.

A.2 Experimental settings

We split the dataset into training data, validation data, and test data by the radio of 0.7:0.1:0.2. We choose Adam Kingma & Ba (2014) as the optimizer in the training phase. During the training phase, the batch size is set to 128 and the epoch size is set to 500, and use RMSprop Xu et al. (2020) for 50 epochs with learning rate as 5^{-4}. The hidden size of GNNs is set to 32, and the hidden state size of LSTMs is set to 64. Since all calculation processes are differentiable, MGST_GNN can be trained via the back propagation method.

We implement our method by PyTorch Paszke et al. (2019), constructing GNNs with PyTorch geometric library Fey & Lenssen (2019) and implement HYSPLIT trajectory analysis and edge weight construction with PySPLIT Warner (2018). The code is released on GitHub[5]. A server with one CPU (Intel®Xeon®Platinum), and one GPU (NVIDIA Tesla T4) accomplish all computing tasks, including training, validation, and test.

We introduce two metrics: MAE (mean absolute error) and SMAPE (symmetric mean absolute percentage error), to evaluate the performances of all methods, which are defined as follows:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |con_i - \widehat{con}_i|}{n}, \tag{9}$$

$$\text{SMAPE} = \frac{1}{n} \sum_{i=1}^{n} \frac{|con_i - \widehat{con}_i|}{(con_i + \widehat{con}_i)/2}, \tag{10}$$

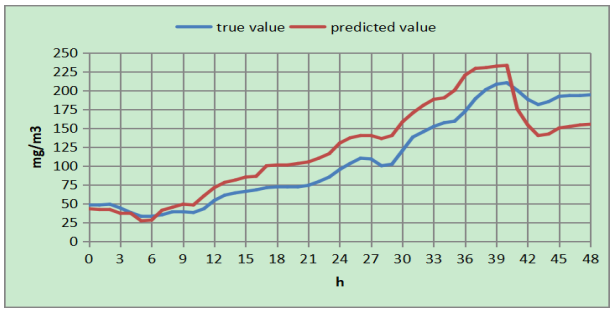
[3]https://lbs.amap.com/api/webservice/guide/api/search/

[4]https://nomads.ncep.noaa.gov/pub/data/

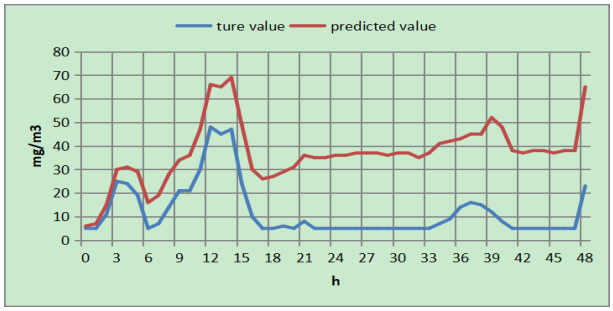
[5]https://github.com/liaohaibing/MGST_GNN/tree/master

Table 3: MAE results of data augmentation evaluation based on JN database

Method	Pollutant category	1h	6h	12h	18h	24h	36h	48h
MGST_GNN	PM2.5	**6.15**	**15.24**	**20.56**	**25.62**	**34.83**	**43.98**	**52.06**
	O_3	**5.07**	**13.37**	**17.16**	**22.19**	**29.92**	**37.96**	**45.27**
w/o DA	PM2.5	6.25	16.97	22.18	28.23	36.96	46.31	54.69
	O_3	5.11	14.84	18.86	23.72	30.78	39.19	48.07



(a)



(b)

Figure 7: 48-hour predicted value versus true value (a) PM2.5 (b) O_3

where con_i and $\widehat{con}_i$ denote the true and forecasted values, respectively.

In the experiments, we utilize previous 48-hour observations and select the results of 1 hour, 6 hours, 12 hours, 18 hours, 24 hours, 36 hours and 48 hours ahead forecasting to report. All experiments are repeated 5 times to avoid contingency.

A.3 Verify the effect of data augmentation

In order to verify the data augmentation effect, in this section, we conduct a comparison experiment between data augmentation and non-data augmentation, as shown in Table 3, where w/o DA indicates that data augmentation technology is not used. In addition, Fig. 7 depicts the predicted PM2.5 and O_3 concentrations and their corresponding real values of a station in Jinan city from 00:00 on January $2nd$, 2021 to 23:00 on January $3rd$, 2021 for consecutive 48 hours.

As seen from Table 3:

1. Using our data augmentation method, the MAE has been significantly reduced;

2. It is easier to predict O_3 concentration than PM2.5, because O_3 has more periodic regularity and stability than PM2.5. Therefore, the subsequent experiments are carried out with the prediction of PM2.5 concentration.

Table 4: Results of factor evaluation based on JN database

Method	Metric	1h	6h	12h	18h	24h	36h	48h
MGST_GNN	MAE	**6.15**	**15.24**	**20.56**	**25.62**	**34.83**	**43.98**	**52.06**
	SMAPE	**0.07**	**0.10**	**0.14**	**0.19**	**0.26**	**0.33**	**0.46**
w/o wdf	MAE	7.06	17.17	23.68	29.33	38.56	46.70	53.78
	SMAPE	0.07	0.11	0.16	0.21	0.28	0.35	0.47
w/o tf	MAE	6.76	17.16	23.56	28.45	34.96	45.55	52.01
	SMAPE	0.07	0.11	0.16	0.20	0.26	0.34	0.46
w/o gf	MAE	6.71	16.78	22.56	28.64	34.95	46.56	52.11
	SMAPE	0.07	0.11	0.15	0.20	0.26	0.35	0.46

Table 5: Results of model component evaluation based on JN database

Method	Metric	1h	6h	12h	18h	24h	36h	48h
MGST_GNN	MAE	**6.15**	**15.24**	**20.56**	**25.62**	**34.83**	**43.98**	**52.06**
	SMAPE	**0.07**	**0.10**	**0.14**	**0.19**	**0.26**	**0.33**	**0.46**
w/o multi granularity	MAE	7.12	17.10	23.59	28.02	34.95	48.02	53.58
	SMAPE	0.08	0.12	0.16	0.21	0.26	0.36	0.47
w/o HYSPLIT	MAE	7.16	18.06	23.59	28.16	34.76	48.35	53.02
	SMAPE	0.08	0.13	0.16	0.21	0.26	0.36	0.47
w/o taLSTM	MAE	6.15	15.85	21.56	28.64	34.96	43.16	52.25
	SMAPE	0.07	0.10	0.15	0.21	0.26	0.33	0.46

A.4 MULTI-SOURCE FACTOR EVALUATION

To verify the effectiveness of multiple factors, we compare MGST_GNN with three variants, each of which removes one kind of factors. Specifically, MGST_GNN w/o wdf removes meteorological forecast data; MGST_GNN w/o tf removes time feature and MGST_GNN w/o gf removes geomorphic feature.

The performances of MGST_GNN and its variants are given in Table 4, as can be seen from the table:

1. MGST_GNN outperforms the other three variants, which indicates that all factors can improve the performance of air quality forecasting;
2. The rank of the effectiveness of factors is weather data>time feature>geomorphic feature. This result shows that air quality is mostly influenced by weather conditions.

A.5 MODEL COMPONENT EVALUATION

To explore the effectiveness of different components, we compare MGST_GNN with three following variants:

1. MGST_GNN w/o multi granularity, which removes the global graph and the corresponding interactions, i.e., the region representation is removed from the global attributes of the partial subgraphs;
2. MGST_GNN w/o HYSPLIT, which removes dynamic edge weight vector by HYSPLIT and use vectors of wind direction and distance instead;
3. MGST_GNN w/o taLSTM, which removes temporal attention mechanism based LSTM at the decoder, and directly use LSTM instead.

The performances of MGST_GNN and the variants mentioned above are given in Table 5, as can be seen from the table:

1. MGST_GNN outperforms MGST_GNN w/o multi granularity in all metrics. The result indicates that the air quality of adjacent regions is beneficial, which can be used to model the diffusion processes of air pollutants from adjacent regions;

Table 6: Model comparison results based on YRD database

Method	Metric	1h	6h	12h	18h	24h	36h	48h
MGST_GNN	MAE	**7.16**	**17.29**	**21.25**	**27.92**	**36.97**	**45.28**	**54.73**
	SMAPE	**0.08**	**0.12**	**0.14**	**0.20**	**0.27**	**0.34**	**0.48**
HighAir	MAE	7.12	17.83	22.65	27.49	38.85	48.36	55.68
	SMAPE	00.08	0.12	0.15	0.20	0.29	0.36	0.49
PM2.5-GNN	MAE	7.15	18.09	23.14	29.57	37.96	48.99	56.12
	SMAPE	0.08	0.13	0.15	0.21	0.28	0.36	0.50
GC-DCRNN	MAE	7.56	18.56	24.88	30.67	43.26	51.84	60.02
	SMAPE	0.08	0.13	0.17	0.22	0.33	0.45	0.53
GC-LSTM	MAE	8.13	18.97	24.53	31.52	42.51	51.98	58.96
	SMAPE	0.09	0.13	0.17	0.22	0.33	0.45	0.52
ST-UNet	MAE	7.22	18.69	23.36	27.86	40.16	51.97	57.85
	SMAPE	0.08	0.13	0.16	0.20	0.32	0.45	0.51
STA-LSTM	MAE	8.15	19.21	24.62	31.96	42.87	50.68	60.37
	SMAPE	0.09	0.14	0.17	0.23	0.33	0.37	0.53

2. MGST_GNN outperforms MGST_GNN w/o HYSPLIT in all metrics. The result indicates that compared with taking wind direction and distance as the edge weight vector, using HYSPLIT to dynamically adjust the weights of edges is a more effective strategy, which can model the effect patterns of wind direction on air pollutant diffusion with domain knowledge;
3. The attention mechanism based LSTM at the decoder is helpful for longer-term prediction, but for short-term prediction.

A.6 Comparison with other forecasting methods

In order to verify the advanced nature of our method, we compare MGST_GNN with the following forecasting methods:

1. HighAir Xu et al. (2021): a hierarchical graph neural network-based air quality forecasting method , which adopts an encoder-decoder architecture and considers complex air quality influencing factors, e.g., weather and land usage;
2. PM2.5-GNN Wang et al. (2020): a graph based model with identify a set of critical domain knowledge for PM2.5 forecasting;
3. GC-DCRNN Lin et al. (2018): it combines recurrent neural networks with diffusion convolution to forecast the air quality. The model describes the spatial relationship by constructing a graph based on the similarity of the built environment among the locations of air quality sensors. To keep the fairness of comparison, we introduce weather data into the input of the decoder;
4. GC-LSTM Qi et al. (2019): this method integrates LSTM and Graph Convolutional Networks (GCN) to model the temporal and spatial dependency respectively. Differing from our MGST_GNN, the GCN module in GC-LSTM only applies to undirected graph, and no edges' attributes are used;
5. ST-UNet Yu et al. (2019): a spatial-temporal forecasting method that leverages pooling operation to coarsen a graph in spatial domain and adopts dilated RNN to capture temporal dependencies;
6. STA-LSTM Zou et al. (2021): a long short-term memory air quality prediction model based on a spatiotemporal attention mechanism.

The performances of MGST_GNN and other forecasting methods are given in Table 6, as can be seen from the table:

1. MGST_GNN outperforms GC-DCRNN, GC-LSTM and ST_UNet, especially in long-term forecasting. It is indicated that a multi-granularity structure can model spatial dependencies more effectively than a flat structure. The reason is that the multi granularity

graph not only considers the local impact of neighboring stations on the prediction station, but also the global impact of different regions on the prediction station, which makes the development of spatial relationships more sufficient;

2. MGST_GNN outperforms HighAir. This is because in the aspect of spatial relationship learning, MGST_GNN uses the professional HYSPLIT model to build edge weight vector. It comprehensively uses meteorological and topographical conditions to calculate the influence relationship among nodes and the specific influence time, making the construction of spatial relationship more accurate and delicate. HighAir simply uses wind direction and distance to construct rough edge weight vector. In terms of temporal relation learning, MGST_GNN adds time feature, and uses the attention mechanism based LSTM in decoder. HighAir simply uses LSTM to build temporal relationships;
3. MGST_GNN outperforms PM2.5-GNN. This is because MGST_GNN uses a multi-granularity spatiotemporal graph network and combines meteorological data, topographic features, time features and professional models to construct node and edge attributes. However, PM2.5-GNN is only a single-granularity spatiotemporal graph network. In the construction of edge attributes, although they make use of domain knowledge, they simply list part of the influencing factors, which is still insufficient. When building node attributes, they only use meteorological data.