



# Self-supervised air quality estimation with graph neural network assistance and attention enhancement

Viet Hung Vu<sup>1</sup> · Duc Long Nguyen<sup>1</sup> · Thanh Hung Nguyen<sup>1</sup> · Quoc Viet Hung Nguyen<sup>2</sup> · Phi Le Nguyen<sup>1</sup> · Thanh Trung Huynh<sup>3</sup>

Received: 17 November 2022 / Accepted: 21 February 2024

© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2024

## Abstract

The rapid progress of industrial development, urbanization, and traffic has caused air quality degradation that negatively affects human health and environmental sustainability, especially in developed countries. However, due to the limited number of sensors available, the air quality index at many locations is not monitored. Therefore, many research, including statistical and machine learning approaches, have been proposed to tackle the problem of estimating air quality value at an arbitrary location. Most of the existing research perform interpolation process based on traditional techniques that leverage distance information. In this work, we propose a novel deep-learning-based model for air quality value estimation. This approach follows the encoder–decoder paradigm, with the encoder and decoder trained separately using different training mechanisms. In the encoder component, we proposed a new self-supervised graph representation learning approach for spatio-temporal data. For the decoder component, we designed a deep interpolation layer that employs two attention mechanisms and a fully connected layer using air quality data at known stations, distance information, and meteorology information at the target point to predict air quality at arbitrary locations. The experimental results demonstrate significant improvements in estimation accuracy achieved by our proposed model compared to state-of-the-art approaches. For the MAE indicator, our model enhances the estimation accuracy from 4.93% to 34.88% on the UK dataset, and from 6.89% to 31.94% regarding the Beijing dataset. In terms of the RMSE, the average improvements of our method on the two datasets are 13.33% and 14.37%, respectively. The statistics for MAPE are 36.05% and 13.25%, while for MDAPE, they are 24.48% and 36.33%, respectively. Furthermore, the value of  $R_2$  score attained by our proposed model also shows considerable improvement, with increases of 5.39% and 32.58% compared to that of comparison benchmarks. Our source code and data are available at <https://github.com/duclong1009/Unsupervised-Air-Quality-Estimation>.

**Keywords** Air quality interpolation · Graph neural network · Time-series prediction

## 1 Introduction

Air pollution and air quality monitoring. Increasing industrialization and urbanization, especially in developing nations, has resulted in severe air pollution in many cities. According to the World Health Organization, air pollution is the leading cause of 36% of lung cancer deaths, 27% of heart attacks, 34% of strokes, and 35% of respiratory failure deaths [1]. In addition, an earlier study [2] has shown a substantial link between air pollution and meteorological variables. Understanding the characteristics of

pollutants is, therefore, essential for dealing with issues related to poor air quality and evaluating the efficacy of environmental initiatives. This need has fostered the worldwide development of air quality monitoring solutions. Traditional air quality surveillance relies on stationary monitoring stations following stringent criteria [3, 4]. The outstanding advantage of this method lies in its high accuracy. Due to the high installation and maintenance expenses, however, there are relatively few stationary monitoring stations constructed, resulting in extremely sparse coverage. To this end, low-cost sensors-based air quality monitoring systems have emerged as a viable alternative to increase the granularity of monitoring [5–8]. However, the sensor-based solution suffers from an inherent limitation of low precision. Indeed, low-cost

---

Viet Hung Vu and Duc Long Nguyen have contributed equally to this work.

---

Extended author information available on the last page of the article

sensor measurements are impacted by several external meteorological factors and have extremely poor accuracy when compared to stationary monitoring stations. Furthermore, although the sensor-based strategy may considerably improve the observation density, covering all regions with sensors is impractical; thus, a comprehensive picture of the geographical distribution of air pollution is often unavailable. In light of this, several efforts have been devoted to proposing a new approach known as *spatial air quality estimation*, which utilizes data acquired from nearby monitored places to forecast air quality at unmonitored locations.

### 1.1 Spatial air quality estimation: existing approaches and their limitations

Unlike the temporal predicting air quality, i.e., utilizing historical air quality data gathered from a particular area to predict the air quality at the same location in future [9–13], the spatial air quality estimation problem has not been substantially studied. The conventional approach to solve the spatial air quality estimation problem is to rely on geostatistical approach including inverse distance weighting (IDW) [14], ordinary kriging (OK) [15], ordinary co-kriging (OCK) [16], land-use regression [17], and Biased Sentinel Hospitals Areal Disease Estimation (STPI-BSHADE) [18]. Such non-learning methods, however, often require domain expertise to engineer special parameters to achieve the optimal solution. Moreover, these techniques cannot model the complicated relationships between air quality indicators and other variables (such as geographical distance and meteorological factors), thereby leading to poor estimation accuracy. Also, the performance of these methods is strongly affected by various factors such as the sampling density and the data variation [16].

Deep learning, which has recently emerged as a viable method for capturing nonlinear and complicated relationships, is being used to tackle a broad range of issues, including classification and regression tasks. Several early attempts have been devoted to utilizing deep learning in estimating air quality indicators spatially, such as [19–21]. Authors in [19] compared the performance of ANN and Multivariate Linear Regression in spatial interpolating five regulated air pollutants (Nitrogen dioxide, Nitrogen monoxide, Ozone, Carbon monoxide, and Sulfur dioxide). The results showed the superiority of ANN in most cases, especially when the density of the monitoring network is limited. In [20], the authors proposed a spatial interpolation/extrapolation method consisting of a geo-layer and LSTM layers. The geo-layer is responsible for selecting stations that strongly correlate to the targeted location, while the LSTM is in charge of deriving temporal correlations from the historical data collected by the selected

stations. Qi et al. in [21] offered an autoencoder-based deep learning model to predict the air quality indicators at unlabeled positions. Li et al. [22] constructed a composite deep learning model that leverages an autoencoder-based full residual and bootstrap aggregation. This model is designed to estimate  $PM_{2.5}$  concentrations with high spatial and temporal precision. Rijal et al. [23] developed an ensemble model of CNN and Feed Forward Neural Network to estimate  $PM_{2.5}$  concentrations using the input of the satellite image dataset.

Despite several attempts, deep learning-based spatial air quality estimation is still in its infancy, with most existing approaches failing to model the spatial relationships of air quality indicators at nearby locations, as well as the correlation between air quality indicators and other external factors (e.g., humidity, temperature, wind speed, wind direction, etc.). Indeed, most estimation methods proposed so far have relied on the inverse distance paradigm to model the spatial relationship and have not considered the correlation between air quality indicators and other external variables. These limitations have hampered estimation accuracy and left this topic as an unsolved challenge.

### 1.2 Our solution

This study focuses on spatial estimation of the  $PM_{2.5}$  indicator (one of the most significant pollutants) and proposes a novel approach for addressing the above-mentioned issues. Specifically, we develop a deep learning model that allows us to model the three essential properties of air quality indicators: (1) spatial correlation between air quality indicators gathered at different places; (2) temporal correlation among air quality indicators obtained from the same location; and (3) relationship between air quality indicators and external meteorological factors.

The main contributions of our paper are fourfold as follows.

- Firstly, we leverage the graph neural network to model the spatial correlation of the  $PM_{2.5}$  indicator collected from various sites. Additionally, we offer a novel adversarial training technique that strengthens the induction capability and noise resistance, thereby enhancing the representation capability of the graph neural network. Moreover, we design a location-based attention mechanism to emphasize significant locations.
- Secondly, we employ recurrent units within the graph neural network to capture the temporal correlation between air quality series gathered from the same site.
- Thirdly, we develop a data fusion mechanism that integrates multiple meteorological data to improve estimation accuracy. Specifically, we introduce a method for representing wind-related information

(considered one of the elements affecting  $PM_{2.5}$  the most) and incorporating it into the input features. Additionally, we develop a feature-based attention mechanism that can simulate the correlations between latent features, thereby highlighting the most significant ones.

- Finally, we conduct extensive experiments on real datasets to justify the performance of our model against existing approaches. The empirical results show that our technique outperforms the others.

The remainder of the paper is organized as follows. Section 2 introduces related works. In Sect. 3, we first formulate the targeted problem and then provide the motivation and overall of our proposed architecture. In Sect. 4, the proposed self-supervised graph representation learning approach is discussed thoroughly, which is followed by detailed implementation of the deep interpolation layer in Sect. 5. In Sect. 6, we detail the experiment settings and raise several questions, which are then answered by experimental results. Finally, we summarize our work in Sect. 7.

## 2 Related works

In this section, we introduce related works about air quality prediction and graph self-supervised representational learning in Sects. 2.1 and 2.2, respectively.

### 2.1 Air quality prediction

Air quality prediction and monitoring have received great interest from both industry and academia in the last two decades. Most of the existing works focused on the setting of predicting the air quality in the near future at the monitoring stations using their historical data [24, 25]. Especially, recent techniques leverage advanced architecture such as recurrent neural network [11, 26] and graph neural network [27, 28] to extract better the underlying pattern from the past data and achieve remarkable prediction accuracy. For instance, Zhao et al. [29] developed a method combining long-short-term memory (LSTM) with fully connected neural network (LSTM-FC) to predict the  $PM_{2.5}$  value of a specific monitoring station over the next 48 h. Qi et al. combined graph convolutional network and LSTM in a novel hybrid model [30] to extract temporal and spatial characteristics of input data. Liang et al. [28] developed a novel method leveraging a multi-level attention mechanism to model the dynamic spatiotemporal dependencies.

However, these methods cannot be applied to estimate the air quality in the unmonitored area, which is essential in

reality due to the shortage of monitoring stations. To predict the air quality indicator at arbitrage areas, the earlier techniques often apply interpolation deterministic methods such as Inverse Distance Weighting (IDW) and Ordinary Kriging (OK) [16]. However, these approaches ignore the impact of historical data, and moreover, they apply a simple linear/nonlinear equation that cannot model the spatiotemporal dynamics of the air quality data. The recent techniques attempt to address this issue by combining the traditional interpolation method with deep learning architecture. For example, in [31], Ma et al. developed a method combining a Bidirectional Long-short-term memory network (BLSTM) with Inverse Distance Weighting (IDW) to fill the areas without monitoring stations. Guo et al. [32] proposed KIDW-TCGRU, which first employs K-nearest Inverted Distance Weighting to generate interpolated data before passing to a Time-Distributed Convolutional Gated Recurrent Unit (TCGRU) model to extract the spatial-temporal characteristics and estimate the air quality. [21] proposed a deep air model to conduct prediction for  $PM_{2.5}$  value in the unsupervised area using the auto-encoder and fully connected network.

Our work goes beyond the existing works by developing an attention-based graph convolution network (GCN) [33] to capture the nonlinear spatial relationship of air quality between geographic locations. We exploit the noise-resistant properties of GCN and augment it with the adversarial training process to enhance the induction capability of the prediction model. We also integrate the rich meteorology information, especially the wind strength and wind direction, to model their dynamic effect on the propagation of air quality from monitored areas to the target location.

### 2.2 Self-supervised graph representation learning

Self-supervised graph representation learning can be classified into three main genres: generation-based method, auxiliary property-based method, and contrast-based method [34]. In the first paradigm, the existing works (e.g., GAE/VGAE [35], MGAE [36], EdgeMask [37]) follow the encoder-decoder method, where the model input for the encoder is the output of the graph perturbation process. The generative decoder tries to reproduce the graph from the encoder output, with a loss aiming to minimize the difference between the original and the reconstructed graphs. The second paradigm, auxiliary property-based methods, has the same training method as the supervised learning paradigm, as both of them require the “sample-label” pair. However, they automatically generate pseudo-labels from several hand-crafted auxiliary graph properties using several unsupervised clustering/partitioning algorithms. Then, the decoder tries to perform classification or regression

using the representation learned from the encoder, with a loss aiming to minimize the difference between the predicted labels and the pseudo-labels. Representative models belonging to this genre are Distance2Cluster, Centrality Score Ranking, Cluster Preserving [38], etc. The disadvantage of this paradigm is the dependence on the accuracy of the pseudo-labels and the selection of auxiliary properties. The third paradigm, contrast-based methods, is built on the idea of mutual information maximization, where the estimated MI between two views of the same object (e.g., node, subgraph, and graph) is maximized, otherwise, minimized. Based on the pretext tasks used, these algorithms can be categorized into two categories: **same-scale** and **cross-scale** contrastive learning. The same-scale approaches such as DeepWalk [39], node2vec [40], GRACE [41], and GraphSAGE [42] use peer instances (e.g., node versus node) to perform discrimination. Cross-scale methods discriminate views across various graph topologies (e.g., node versus graph). For example, DGI [43] learns the contrast between graph-label representation and node-level representation. STDGI [44] improved DGI by analyzing spatiotemporal graphs and maximizing the mutual information between node features at the adjacent time steps.

### 3 Problem formulation and approach

In this section, we first give a formulation for the spatial air quality estimation problem in Sect. 3.1. We then discuss the design principle and the overview of our approach in Sect. 3.4.

#### 3.1 Problem formulation

**Monitoring station grid.** We assume that there are  $n$  monitoring stations located at different locations, forming a station grid  $\mathbb{S}$ . Each monitoring station  $S_i \in \mathbb{S}$  in the grid is associated with a coordinate  $C_i = (\varphi_i, \lambda_i)$ , where  $\varphi_i, \lambda_i$  represent the latitude and longitude, respectively. We denote by  $D(S_i, S_j)$  the geographical distance between two stations  $S_i, S_j \in \mathbb{S}$ . In this study, the Haversine function [45], a popular method for calculating the sphere distance, is employed. We constraint that there exists a maximum distance threshold  $D^*$  such that for every station  $S_i$ , there exists at least one station  $S_j$  staying within its radius  $D^*$ , i.e.,  $D(S_i, S_j) < D^*$ . For real-world setting,  $D^*$  often does not exceed 200 km [30]. Regarding the outlined restriction, our proposed method is best suited for country-level application.

#### 3.2 Multivariate air quality data

The interpolation of air quality in the unmonitored area relies on data collected from the nearby monitoring stations. The monitoring stations gather two kinds of data: air quality indicators (e.g.,  $\text{PM}_{2.5}$ ,  $\text{CO}_2$ ,  $\text{NO}_2$ , etc.) and meteorological data (e.g., temperature, evaporation, wind speed, precipitation, etc.).

The multivariate historical data at the station  $S_i$ , denoted as  $\mathbf{X}_i$ , have the form of  $\mathbf{X}_i = \{X_i^0 \dots X_i^{t-1}, X_i^t, \dots X_i^T\}$ , with  $X_i^t = (Q_i^t, M_i^t)$  where  $T$  is the current timestamp, and vector  $Q_i^t, M_i^t$  represent air quality indicators and meteorological data collected at location  $S_i$  at timestamp  $t$ , respectively. Furthermore, we assume that  $Q_i^t = \{q_{i-0}^t, q_{i-1}^t, \dots, q_{i-m}^t\}$ , where  $q_{i-0}^t$  depicts the target air quality indicator, e.g.,  $\text{PM}_{2.5}$ , and  $q_{i-1}^t, \dots, q_{i-m}^t$  represent other indicators. Specifically, the air quality vector  $Q_i^t$  contains information on  $\text{PM}_{2.5}$ , AQI,  $\text{PM}_{10}$ , CO,  $\text{NO}_2$ , and  $\text{O}_3$  for the Beijing dataset, and  $\text{PM}_{2.5}$ ,  $\text{PM}_{10}$ ,  $\text{O}_3$ ,  $\text{NO}_2$ , and  $\text{SO}_2$  for the UK dataset. The meteorological vector  $M_i^t$  includes Temperature, Surface pressure, Evaporation, and Precipitation data for both datasets. The length of vector  $Q_i^t$  varies between 6 for the Beijing dataset and 5 for the UK dataset, while vector  $M_i^t$  has a constant length of 4 for both datasets. Since vector  $X_i^t$  is the concatenation of vectors  $Q_i^t$  and  $M_i^t$ , its length is 10 for the Beijing dataset and 9 for the UK dataset.

#### 3.3 Problem formulation

Given a monitor station grid  $\mathbb{S} = \{S_1, \dots, S_n\}$  with the historical multivariate data of  $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\}$ , a target location  $S$  satisfying the distance constraint of  $D(S, S_i) < D^* (\forall i = 1, \dots, n)$ , our objective is to estimate the current  $\text{PM}_{2.5}$  indicator at  $S$ . We assume that meteorological data (e.g., temperature, evaporation, wind speed, and precipitation) are available for any region, including the target location  $S$ . This information can be easily collected using publicly available sources such as Copernicus [46]. We denote the vector representing meteorological data at the target location at time step  $t$  as  $\mathbf{M}^t$ . Furthermore, we also have the detailed coordinate of the target location, denoted by  $C$ , as well as those of the monitoring stations, denoted by  $\mathbf{C} = \{C_1, \dots, C_n\}$ .

The aim of this study is to develop a predictive model, denoted as  $\mathcal{P}$ , which takes into account historical data from monitoring stations, meteorological data from the target location, and the coordinates of both the monitoring stations and the target location. The model is designed to provide an estimation of the  $\text{PM}_{2.5}$  indicator at the target location as its output.

The problem can be mathematically represented as follows.

**Input:**

$\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\}$ : Historical data at the monitoring stations.

$\mathbf{C} = \{C_1, \dots, C_n\}$ : Coordinates of the monitoring stations.

$\mathbf{M}^T$ : Meteorological data at the target location at the current time step.

$C$ : Coordinate of the target location.

**Output:**  $p = \mathcal{P}(\mathbf{X}, \mathbf{C}, \mathbf{M}^T, C)$ ,

where  $p$  is the estimated  $\text{PM}_{2.5}$  indicator at target location  $S$  at current time step.

The notations and operators used in the paper are summarized in Tables 1 and 2, respectively.

### 3.4 Design principle

We argue that a framework tackling the above problem should overcome the following challenges:

- *C1: Spatiotemporal dependency*: Fine-grained air quality maintains both temporal and spatial dependency [47], which means that current air quality indicators are often relevant to its historical data as well as the air quality indicators at nearby locations. The challenge

**Table 2** Important operations

Operator	Definition
$\cdot$	Dot product
$\odot$	Hadamard product
$[\cdot]$	Concatenation

here is to design an architecture that is capable of capturing both these information at the same time.

- *C2: Multi-modal information*: The air quality data consist of multivariate features, including different air quality indicators and meteorology features. Some features might impact the prediction of  $\text{PM}_{2.5}$  indicator more than others. Thus, the challenge is how to effectively integrate this varied range of features into the model to boost the accuracy of the target feature indicator estimation.
- *C3: Interpolation capability*: In the problem of estimating air quality at an arbitrary unknown location, the lack of historical air quality data raises a significant challenge. To overcome this issue, the framework should be able to model the correlation between the locations based on available stations and generalize to arbitrage places. The direct application of existing interpolation technique such as Inverse Distance Weighting (IDW) may not fully model the non-Euclidean characteristics of the input data.

**Table 1** Important notations

Symbols	Definition
$S_i$	$i$ -th monitoring station
$C_i = (\phi_i, \lambda_i)$	coordinates of $S_i$
$S$	target location
$C = (\phi, \lambda)$	the coordinate of $S$
$\mathbb{S}$	monitoring station grid
$\mathbf{C}$	coordinates of monitoring stations
$n$	the number of monitoring stations
$Q_i^t$	air quality indicators collected by $S_i$ at time step $t$
$M_i^t$	meteorological information collected by $S_i$ at time step $t$
$X_i^t = (Q_i^t, M_i^t)$	air quality and meteorology information collected by $S_i$ at time step $t$
$\mathbf{X}_i$	historical data collected by $S_i$ up to the current time step
$\mathbf{X}^t$	historical data collected by all monitoring stations at time step $t$
$\mathbf{X}$	historical data collected by all monitoring stations up to the current time step
$\mathbf{M}^T$	meteorology information at target location at the current time step $T$
$\mathbf{S}^T$	wind score vector at target location at the current time step $T$
$A$	monitoring stations' adjacency matrix
$G^t = (\mathbf{X}^t, A)$	the graph representing information of all monitoring stations at time step $t$
$D(S_i, S_j)$	distance between $S_i$ and $S_j$
$\mathbf{D}$	distances between all monitoring stations' locations and the target location
$p$	estimated $\text{PM}_{2.5}$ level at target location $S$



### 3.5 Framework overview

In this work, we aim to address the above-mentioned challenges by developing a framework that performs interpolation at any specified location. To address **C1**, we propose a temporal graph convolution network (T-GCN) that can learn the spatiotemporal dynamic of the input data. T-GCN comprises of Gated Recurrent Network (GRU) [48], which is well-known for its strong capability in handling sequence data, and Graph Convolutional Network [33], a deep network that can effectively capture the relationship between nodes in spatial domain using node feature. We then propose a novel approach using the contrastive learning paradigm, in which we design a corrupt function that utilizes both the global view corruption and feature-level corruption. This learning mechanism aims to increase the induction capability of the proposed network. Aiming to solve the problem **C2**, we propose a preprocessing process to analyze wind direction from the neighboring stations to the target location, which is then further used to modify the current wind strength based on its directness to the location of interest. Besides, each input feature has a different influence on the target feature. Regarding this challenge, we offer a feature-aware attention mechanism that adaptively scores the essential factor of each feature and highlights the relevant features. Concerning **C3**, we develop a deep-learning-based interpolation method that leverages the location-aware attention mechanism to learn the inter-station dependency. Furthermore, this technique also considers the meteorology features at the target point as an additional input feature to further enhance the performance of the proposed model.

To realize the functions discussed above, we design the framework following the self-supervised training paradigm as shown in Fig. 1. In particular, our model is comprised of three major components: **Multivariate input data representation**, **Graph-based Spatiotemporal modeling**, and **Attention-based interpolation**.

- *Multivariate input data representation*: The main objective of this block is to receive the input data and organize it into the graph-based structure. Specifically, the input data consist of historical data acquired from the monitoring network, coordinates of the monitoring stations and the location of interest, meteorological information at the target location, and distances between the target location and the monitoring stations. Specifically, the data of the monitoring network consist of the historical data from the previous  $k$  time-step to the current time-step, which is represented by  $k$  graphs,  $\{G^{T-k+1}, \dots, G^T\}$ . In addition, each graph  $G^t$  ( $t = T - k + 1, \dots, T$ ) is represented by  $(\mathbf{X}^t, \mathbf{A})$ , where  $\mathbf{X}^t$  is the attribute matrix (representing the data

collected at all stations at time step  $t$ ), and  $\mathbf{A}$  is the adjacency matrix representing the inverse-distances of the stations. The details are provided in Sect. 4.1.

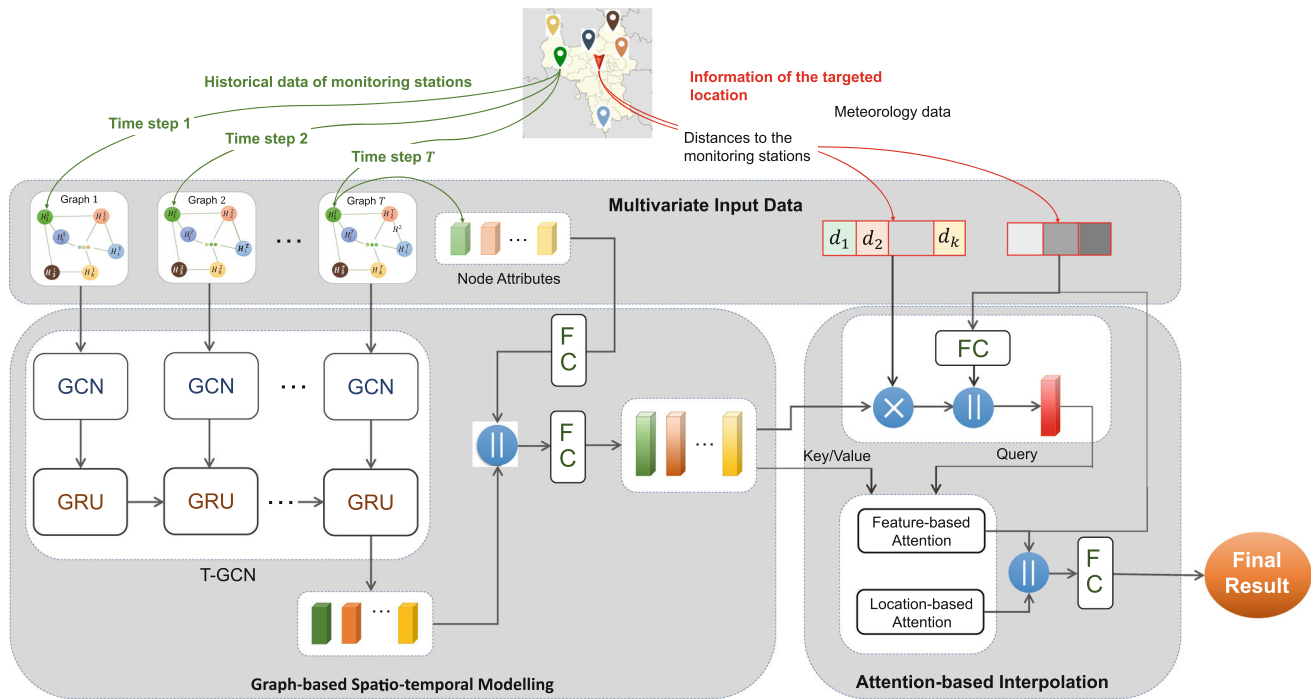
- *Graph-based Spatiotemporal modeling*: This component is responsible for modeling the spatial-temporal correlation of the data obtained from the multivariate input data representation block and transforming it into the latent vector space. Specifically, the graph-based spatiotemporal modeling block comprises multiple graph neural networks and a recurrent neural network. The former captures the spatial relationship between the monitoring stations, while the latter extracts the sequential features of the air quality data series across multiple time steps. The features retrieved by the graph-based spatiotemporal modeling block are then fused with the information of the target location (i.e., meteorological and distance data) before being passed to the Attention-based interpolation module. This procedure is described in detail in Sect. 4.
- *Attention-based interpolation*: This module acts as a decoder that receives information from the graph-based spatiotemporal modeling block, applying two attention mechanisms to highlight important information and produce the final prediction results. In the Attention-based interpolation module, we employ two attention mechanisms, namely location- and feature-based attention, to capture the correlation between the locations and latent features and to highlight the most significant ones. The output of the Attention-based interpolation module is the final estimation result. The details of this block will be discussed in Sect. 5.

## 4 Spatiotemporal graph representation learning

In this section, we first describe the construction of the spatiotemporal input graph in Sect. 4.1. After that, we go through the structure of the spatiotemporal embedding network in Sect. 4.2. Finally, we elaborate on the process of training the embedding network following the contrastive learning paradigm in Sect. 4.5.

### 4.1 Construction of the input graph network

The historical data of the monitoring stations in  $k$  time steps (from  $T - k + 1$  to  $T$ ) are represented by  $k$  graphs. Specifically, at each time step  $t$  ( $T - k + 1 \leq t \leq T$ ), we construct a complete weighted graph  $G^t$  whose nodes represent the monitoring stations, and the weight of each edge reflects the correlation between the two vertices. As mentioned in Sect. 3.5,  $G^t$  can be represented as



**Fig. 1** Overview of our proposed framework, which is comprised of three major components: multivariate input data representation, graph-based spatiotemporal modeling, and attention-based interpolation

$G^t = \{\mathbf{X}^t, A\}$ , where  $\mathbf{X}^t \in R^{n \times l}$  conveys air quality and meteorological data of all monitoring station at time step  $t$ , and  $A \in R^{n \times n}$  is the adjacency matrix,  $n$  is the number of monitoring stations, and  $l$  is the total number of air quality indicators and meteorological features. As the correlation between two stations' air quality measurements tends to be inversely related to their geographical distance [49], we utilize the geographical distance to define the weighted adjacency matrix as follows.

$$A_{ij} = \begin{cases} \frac{1}{D(S_i, S_j)} & \text{if } D(S_i, S_j) > 1 \\ 1 & \text{if } D(S_i, S_j) \leq 1, \end{cases} \quad (1)$$

where  $D(S_i, S_j)$  is the geographical distance, measured in kilometers, between two stations  $S_i$  and  $S_j$ . Note that when the distance is extremely small, its inverse value tends to approach infinity. Therefore, to avoid the situation where some elements of the adjacency become excessively large, we define a geographical distance threshold, for every station pair whose distance is below this lower bound, the weight of their connecting edge is set to a constant. In this paper, we set the threshold and the constant to 1. It is important to note that the same distance constraint, as described in [30], is applied to the stations in the network. Specifically, we only consider stations with adjacent stations located within a maximum distance of 200 kms.

Concerning the node features  $\mathbf{X}^t$ , besides the common scalar factors used in existing works such as  $\text{NO}_2$ ,  $\text{O}_3$ ,

temperature, and precipitation, we also introduce the wind-related information as this is an important factor that affects the air quality indicators, especially  $\text{PM}_{2.5}$  [50]. For instance, air pollution would be more significant when the wind blows directly from the contaminated source to the target area. However, the wind vector, including the strength and direction, is often unavailable in the monitoring stations. Therefore, we propose a method to measure the influence of the wind blowing from a monitoring station to the targeted location, considering the wind direction.

Let  $C = (\lambda, \varphi)$ , and  $C_i = (\lambda_i, \varphi_i)$  be the coordinates of the targeted location and the monitoring station  $S_i$ , respectively. Using the Haversine formula, we determine the angle between  $S_i$  and  $S$ , denoted as  $\theta_i$  as follows.

$$\begin{aligned} \text{hav}(\theta_i) &= \text{hav}(\varphi - \varphi_i) + \cos(\varphi_i) \cos(\varphi) \text{hav}(\lambda - \lambda_i), \\ \text{hav}(\theta_i) &= \sin^2\left(\frac{\theta_i}{2}\right) \end{aligned} \quad (2)$$

Intuitively, the smaller  $\theta_i$  is, the more impact the wind from  $S_i$  imposes on the targeted location. Motivated by this intuition, we define a so-called *wind score*  $s_i$  of a monitoring station  $S_i$  to the targeted location as follows.

$$s_i = \begin{cases} \cos(\theta_i) & \text{if } 0^\circ < \theta_i < 90^\circ \\ 0 & \text{if } \theta_i > 90^\circ, \end{cases} \quad (3)$$

This hand-crafted feature is then combined with other air quality indicators and meteorological data to form the node attribute in the input graphs.

## 4.2 Spatiotemporal embedding network

We employ a deep network architecture onto the constructed graphs to learn simultaneously spatial and temporal properties from the input data. The spatiotemporal network consists of two primary components: a *spatial extractor* which leverages graph neural network to capture spatial correlation patterns from the input stations; and a *temporal extractor* which utilizes the recurrent neural network to encode temporal correlation.

### 4.2.1 Spatial extractor

Aiming to capture the spatial relation among monitoring stations, we employ a deep neural network architecture that has a strong ability to effectively describe the graph structure of the real-world dataset and to capture the spatial dependency between data points in various locations. Hence, we propose using GCN [33], a multi-layer graph neural network that applies a neighborhood aggregation scheme  $f(\cdot)$  on each layer. Considering the  $k$ -layer GCN network, the feed-forward pass process is:  $f^{(k)}(f^{(k-1)}(\dots f^{(1)}(G)\dots))$ , where  $G$  is the input network. The function  $f(\cdot)$  takes the hidden features of the previous layer as input and produces the features of the subsequent layer:

$$\mathbf{H}^{l+1} = f(G, \mathbf{H}^l, W^l) = \sigma(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} \mathbf{H}^l W^l), \quad (4)$$

where  $\hat{A} = A + I_n$  is the adjacency matrix of the with added self-connections,  $I_n$  is the identity matrix,  $\hat{D}$  is the diagonal matrix representing the node degree of the adjacency matrix  $A$ , and  $W^l$  represents the weight matrix in the  $l$ -th layer,  $\mathbf{H}^l$  is the  $l$ -th layer output embedding with  $\mathbf{H}^0$  is the matrix comprised of the attributes of all nodes,  $\sigma$  denotes the activation function, which is ReLU in our proposed approach. Aiming to balance between the expressive power and the computation efficiency of the model, we choose the number of network layers  $k$  equals 2.

### 4.2.2 Temporal extractor

We leverage GRU network [48] to extract temporal characteristics of data series collected from monitoring stations in multiple time steps. GRU is chosen due to its simplicity and capability in dealing with gradient-related problems (i.e., gradient vanishing, gradient explosion).

Inspired by T-GCN [51], we combine the two extractors in the following manner to capture the spatiotemporal

dynamics. First, each input graph  $G^t$  (carrying information about the monitoring station grid at time step  $t$ ) is fed to a GCN unit to extract the temporal correlation between the stations at that time step. The output of each GCN unit at time step  $t$ , along with the output of the GRU unit at the previous time step, is then sent into the GRU unit to capture the temporal properties until time step  $t$ . Finally, the output of the last GRU unit is fused with the node attributes of the last input graph (i.e.,  $G^T$ ) to generate latent vectors encapsulating spatiotemporal information of all historical data acquired from the monitoring station grid. More specifically, the operation inside each GRU cell at time step  $t$  is defined as follows:

$$\begin{aligned} r_t &= \sigma(W_r f_{gc}(\mathbf{X}^t, A) + U_r h_{(t-1)} + b_r), \\ z_t &= \sigma(W_z f_{gc}(\mathbf{X}^t, A) + U_z h_{(t-1)} + b_z), \\ c_t &= \tanh(W_c f_{gc}(\mathbf{X}^t, A) + h_{(t-1)}(R_t \odot W_n) + b_n), \\ h_t &= (1 - z_t) \odot c_t + z_t \odot h_{(t-1)}, \end{aligned} \quad (5)$$

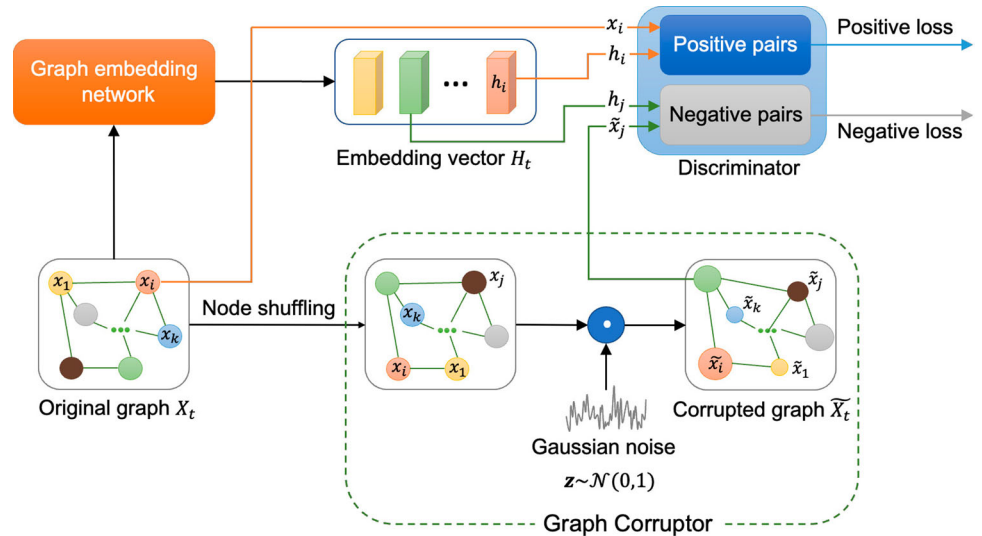
where  $W$  and  $U$  denote the weight matrices for each control gate and the  $b$  terms are bias vectors,  $\odot$  depicts the Hadamard product,  $\sigma$  represents the activation function, and  $r_t$ ,  $z_t$ ,  $c_t$ , and  $h_t$  are the reset gate, update gate, candidate hidden state, and hidden state of the GRU unit, respectively.

## 4.3 Adversarial self-supervised training

We design an adversarial self-supervised learning mechanism (see Fig. 2) to train the embedding network efficiently. Besides the embedding network, our mechanism consists of two components: the *graph corruptor* and the *discriminator*. The former is responsible for tweaking the original graph to generate a negative sample, while the latter distinguishes between the positive and negative examples. Our training flow is performed as follows. Each training sample  $\mathbf{X}^t = \{x_1^t, \dots, x_n^t\}$  (i.e., the graph representing information of all monitoring stations at time step  $t$ ) is fed into the graph embedding network, which generates embedding vectors  $\mathbf{H}^t = \{h_1^t, \dots, h_n^t\}$  with  $h_i^t$  representing information about node  $x_i^t$ . On the other hand, the original graph  $\mathbf{X}^t$  is also corrupted by the Graph Corruptor to yield  $\tilde{\mathbf{X}}^t$ . Every pair of an embedding vector  $h_i^t$  and its corresponding node in the original graph, i.e.,  $x_i^t$ , is considered positive, whereas a pair of an embedding  $h_j^t$  and its corresponding node in the corrupted graph, i.e.,  $\tilde{x}_j^t$  is considered negative. The graph embedding network and discriminator are trained adversarially to minimize the loss associated with positive pairs and maximize the loss contributed by negative pairs.



**Fig. 2** An illustration of our proposed self-supervised training process, which consists of two components: the graph corruptor and the discriminator. The former is responsible for tweaking the original graph and generating a negative sample, while the latter is for distinguishing between positive and negative pairs



In the following, we will provide the details of the graph corruptor and the discriminator.

#### 4.3.1 Graph corruptor

The objective of the corruptor is to manipulate the original graph in order to generate a so-called negative sample that is distinct from the original. Notably, the more diverse the corrupted graphs, the more generalized the data used to train the graph embedding network. Therefore, we design the graph corruptor module that augments the original graph at both structure and attribute levels.

For the structural perturbation, we apply a row-wise shuffling process as in [44]. Since we randomly swap the value of the node feature, this process equals changing the topology structure of the graph. Then, we apply attribute corruption by adding a Gaussian noise to each node feature. More specifically, we sample the random vector  $\tilde{m} \in \mathbb{R}^l$ , where  $l$  is the number of node attributes. Each element of this vector is drawn from a Gaussian distribution of range  $[0; 1]$ . Then, the corrupted node features  $\tilde{\mathbf{X}}$  are computed by:

$$\tilde{\mathbf{X}} = [x_1 \odot \tilde{m}, x_2 \odot \tilde{m}, \dots, x_n \odot \tilde{m}]^T, \quad (6)$$

where  $\odot$  is the Hadamard product operator.

#### 4.3.2 Discriminator

Based on the concept of mutual information (MI) maximization [44], the discriminator is trained such that the mutual information between each node embedding and its respective node feature is maximized and is minimized otherwise. Let  $\mathbf{X}$  be an input graph, and suppose  $\mathbf{X} = \{x_1, \dots, x_n\}$ . Besides, let us denote by  $\tilde{\mathbf{X}} = \{\tilde{x}_1, \dots, \tilde{x}_n\}$ , and  $\mathbf{H} = \{h_1, \dots, h_n\}$  be the corrupted version and the

embedding vector of  $\mathbf{X}$ . We generate several positive pairs  $(h_i, x_i)$ , and negative pairs  $(h_j, \tilde{x}_j)$  ( $1 \leq i, j \leq n$ ). For each  $(h_i, x_i)$ , the discriminator  $\mathcal{D}$  predicts the probability for  $(h_i, x_i)$  being a positive pair by applying a bilinear scoring function.

$$\mathcal{D}(x_i, h_i) = \sigma(h_i^T W x_i). \quad (7)$$

Similarly, the probability for a negative pair  $(h_j^t, \tilde{x}_j^t)$  being a positive pair is defined by

$$\mathcal{D}(\tilde{x}_j, h_j) = \sigma(h_j^T W \tilde{x}_j), \quad (8)$$

where  $W$  is a learnable matrix, and  $\sigma$  is the logistic sigmoid activation function. We design a contrastive loss  $\mathcal{L}_{ssl}$  as follows.

$$\mathcal{L}_{ssl} = \frac{1}{2N} \left( \sum_{i=1}^N \mathbb{E}_{(\mathbf{x}, \mathbf{A})} [\log \mathcal{D}(x_i, h_i)] + \sum_{j=1}^N \mathbb{E}_{(\tilde{\mathbf{x}}, \tilde{\mathbf{A}})} [\log (1 - \mathcal{D}(\tilde{x}_j, h_j))] \right),$$

where  $N$  is the number of the positive and negative samples. The encoder is then trained separately following this loss function and updates the weight of parameters after each epoch using the Adam optimizer [52].

## 5 Multi-level attention interpolator

To interpolate the air quality at unmonitored areas using historical time-series collected from the monitoring station grid, we design a multi-level attention mechanism consisting of two attention layers, namely *location-aware attention* and *feature-aware attention*. The former is responsible for capturing inter-station relationships and emphasizing the most significant stations to the location of interest. In the meantime, the latter is able to model the

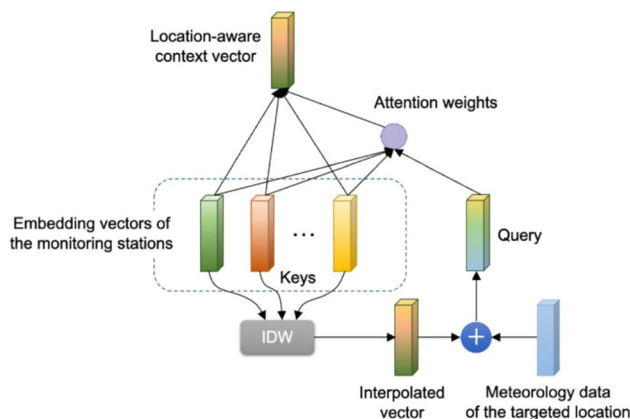
relationship within latent characteristics that pertain to the same locations.

### 5.1 Location-aware attention

After passing input graphs through the graph-based spatiotemporal modeling module, we acquire embedding vectors, each of which retrieves information related to a monitoring station. Obviously, the correlation of each monitoring with the target location is different. Consequently, the contribution of each monitoring method station in determining  $\text{PM}_{2.5}$  at the target station is also distinct. Typically, nearby stations have a more significant impact on the targeted site than distant stations. However, these effects vary throughout time and are influenced by numerous circumstances. For instance, if a strong wind blows from a monitoring station to the target location, the air quality values acquired at the monitoring station may exhibit a substantial association with those at the target location. Inspired by this observation, we design a new attention mechanism that highlights the most relevant monitoring stations. The architecture of this mechanism is depicted in Fig. 3. Specifically, we first create a so-called *interpolated embedding vector*  $h$ , which is defined by the weighted sum of embedding vectors obtained from the graph modeling module as follows.

$$h = \frac{1}{\sum_{k=1}^n d_k^{-1}} \sum_{j=1}^n d_j^{-1} h_j, \quad (9)$$

where  $h_j$  is the embedding vector corresponding to the monitoring station  $S_j$ , and  $d_j$  is the distance between the target location  $S$  and monitoring stations  $S_j$ , and  $n$  is the number of the monitoring stations. Intuitively, this  $h$  can be seen as a rough estimation of the air quality at the targeted location using the information from monitoring stations.



**Fig. 3** An illustration of our location-aware attention mechanism, which highlights monitoring stations with the highest correlation to the target site

The interpolated embedding vector  $h$  is then combined with the meteorological data at the target location to form the query, while the embedding vectors of the monitoring stations are employed as the keys and values for the attention block. For each key  $h_i$ , its attention weight  $g_i$  is defined as follows.

$$g_i = W_g h_i \cdot U_g [h; m], \quad (10)$$

$$m = W'_g [\mathbf{M}^T; \mathbf{S}^T],$$

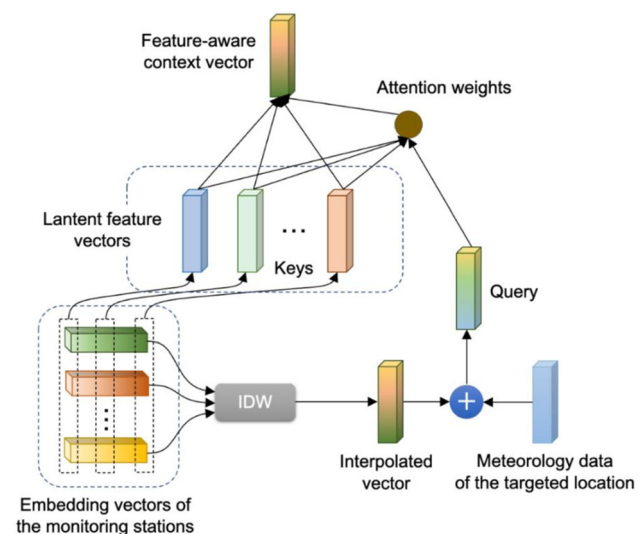
where  $W_g, U_g, W'_g$  are the learnable parameters,  $\mathbf{M}^T, \mathbf{S}^T$  are the meteorological data and the wind score the at the targeted location at current time step  $T$ ,  $h$  is the interpolated feature vector. Finally, the attention weights  $g_i$  are normalized and employed to calculate the final context vector as follows.

$$\beta_i = \frac{\exp(g_i)}{\sum_{j=1}^n \exp(g_j)}, \quad (11)$$

$$\tilde{h}_{\text{location}} = \sum_{i=1}^n \beta_i \times h_i.$$

### 5.2 Feature-aware attention

As described in the previous section, the output of the graph-based spatiotemporal modeling block is latent vectors. However, not every component in these latent vectors contributes equally to the prediction of the  $\text{PM}_{2.5}$  indicator. To reduce the impact of irrelevant components and emphasize those of highly correlated components, we design a so-called feature-aware attention mechanism as shown in Fig. 4. The feature-aware attention uses the same



**Fig. 4** An illustration of our proposed feature-aware attention mechanism, which highlights the most significant latent features

query as the location-aware attention, i.e., the concatenation of the interpolated feature vector and the meteorological data of the targeted location. On the other hand, there are  $n$  keys/values that are created by concatenating components from embedding vectors produced by the graph-based spatiotemporal modeling block. Specifically, the  $j$ -th key is the combination of the  $j$ -th latent features of embedding vectors  $h_1, \dots, h_n$ . Let  $k_j$  be the  $j$ -th key; then, its attention weight is defined as follows.

$$\begin{aligned} e_j &= W_f k_j \cdot U_f [h; m], \\ m &= W'_f [\mathbf{M}^T; \mathbf{S}^T], \end{aligned} \quad (12)$$

where  $W_f, U_f, W'_f$  are the learnable parameters and  $\mathbf{M}^T, \mathbf{S}^T$  are the meteorology feature vector, and the wind score vector at the target location at the current time step  $T$ , respectively. Finally, the attention weights are normalized and utilized to calculate a context vector as follows.

$$\begin{aligned} \gamma_j &= \frac{\exp(e_j)}{\sum_{j=1}^m \exp(e_j)}, \\ \tilde{h}_{\text{feature}} &= \sum_{j=1}^m \gamma_j \times k_j. \end{aligned} \quad (13)$$

### 5.3 Air quality interpolator training

Given the context vectors generated by the attention blocks, we will now present the last step in our model that generates the estimation result. We observe that meteorological factors, including precipitation, temperature, evaporation, and wind, play a substantial effect in determining  $\text{PM}_{2.5}$  indicator. Therefore, we propose integrating these meteorological data and the wind score vector at the target location with the context vectors via a fully connected layer. The output of our model can be represented mathematically as follows.

$$\mathbf{O} = FC(\sigma(FC(W_i[\tilde{h}_{\text{location}}; \tilde{h}_{\text{feature}}; \mathbf{M}^T; \mathbf{S}^T]))W_o), \quad (14)$$

where  $W_i, W_o$  is the learnable matrices,  $\sigma$  is the ReLU activation function.

We leverage the MSE (Mean Square Error) loss function to train the model whose formula is as follows.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \tilde{y}_i)^2, \quad (15)$$

where  $y_i$  and  $\tilde{y}_i$  are the estimation result and the ground truth, respectively, and  $N$  is the number of observations in the input data batch. The overall training process of our framework is summarized in Algorithm 1.

#### Algorithm 1 Training process

---

**Input:** input feature matrix  $\mathbf{X}$ , list of stations' coordinates  $\mathbf{C}$ , target coordinate  $C$ , meteorology information at the target location  $\mathbf{M}^T$ , encoder  $\mathcal{E}$ , corruptor  $\mathcal{C}$ , discriminator  $\mathcal{D}$

**Output:** Current  $\text{PM}_{2.5}$  indicator at the target location  $\mathbf{O}$

Compute the adjacency matrix  $\mathbf{A}$  following Equation 1

**repeat**

- Calculate the input feature embedding:  $\mathbf{H} = \mathcal{E}(\mathbf{X})$
- Retrieve corrupted input  $\tilde{\mathbf{X}} = \mathcal{C}(\mathbf{X})$
- Compute the probability of each sample pair being positive following Equation 7
- Compute loss  $\mathcal{L}_{ssl}$  following Equation 8
- Update the encoder parameters

**until** reach convergence;

Frozen parameters of the encoder network

Calculate the list of distances  $\mathbf{D}$  using Haversine formula

**repeat**

- Calculate interpolated embedding vector  $h$  following Equation 9
- Compute context vectors  $\tilde{h}_{\text{feature}}$  and  $\tilde{h}_{\text{location}}$  using the Equation 13 and 11
- Compute  $\mathbf{O}$  following Equation 14
- Compute the  $MSE$  loss following the Equation 15
- Update the decoder parameters

**until** reach convergence;

---

## 5.4 Air quality interpolation process for unsupervised places

Given the trained model, the historical data from the monitoring stations, the meteorological data at the target location, and the distances from the target location to the monitoring stations, we then perform the interpolation process. It is important to highlight that during the training process of the encoder, we use all input features, including PM-related features, such as PM<sub>2.5</sub>, PM<sub>10</sub>, and AQI. Conversely, during the training of the interpolator, we restrict our focus to meteorological features at the target location (e.g., Temperature, Precipitation, Surface pressure) and historical input features from neighboring stations. This training scheme aligns with methodologies employed in related works [31, 32]. Firstly, the historical data of the monitoring stations are fed into the graph-based spatiotemporal modeling block to extract embedding vectors. These embedding vectors are then combined with the meteorology information at the target location and passed through two attention blocks to generate the context feature- and location-aware context vectors. Finally, the context vectors are fused with the meteorological data and the wind score vector at the target location by a fully connected layer to produce the final prediction result. The whole inference process is summarized in Algorithm 2.

**Algorithm 2** Testing process

---

**Input:** input feature matrix  $\mathbf{X}$ , list of stations' coordinates  $\mathbf{C}$ , target coordinate  $C$ , current meteorology information at the target location  $\mathbf{M}^T$ , current wind impact score vector at the target location  $\mathbf{S}^T$ , encoder  $\mathcal{E}$

**Output:** current air quality value at target location  $\mathbf{O}$

Compute the adjacency matrix  $\mathbf{A}$  following Equation 1

Compute input feature embedding  $\mathbf{H} = \mathcal{E}(\mathbf{X})$

Re-calculate the list of distances  $\mathbf{D}$  using Haversine formula [46]

Calculate interpolated embedding vector  $h$  following Equation 9

Compute context vectors  $\tilde{h}_{\text{feature}}$  and  $\tilde{h}_{\text{location}}$  using the Equation 13 and 11

Compute  $\mathbf{O}$  following Equation 14

---

## 6 Performance evaluation

In this section, we conduct experiments with the aim of answering the following research questions:

- (RQ1) Does our proposed model outperform the baseline methods?
- (RQ2) How important is each design choice affecting our model?
- (RQ3) How important is each input feature to the estimation of air quality?

- (RQ4) Is our technique interpretable?

In the following, we first describe the dataset we used throughout the experiment in 6.1, and the experimental setting in 6.3. We then conduct our empirical evaluations to answer the stated questions, including an end-to-end comparison in 6.5, an ablation study in 6.6, a feature importance analysis in 6.7, and an analysis of the relationship between distance and the predictive performance in 6.10.

### 6.1 Study area and experimental setup

*Study area* We conducted our study in two distinct geographical locations: Beijing, China, and the United Kingdom. Further details regarding the specifics of each dataset are provided below.

- *UK Dataset* In 2021, Reani et al [53] published a dataset of UK daily meteorology, air quality, and pollen measurements for four consecutive years from 2016 to 2019. This dataset covers an area of 242,295  $\text{km}^2$ , including varied kinds of topography consisting of rugged, undeveloped hills and low mountains, and rolling plains. The authors collected daily data of temperature, evaporation, precipitation, wind speed, O<sub>3</sub>, NO<sub>2</sub>, SO<sub>2</sub>, PM<sub>10</sub> and PM<sub>2.5</sub>, over the period of 1462 days. The dataset provides data from 141 air quality stations across the United Kingdom.

- *Beijing Dataset* The Beijing [54] dataset collects the air quality and meteorological information of 35 stations across Beijing in 2018, which includes 8643 data points. This dataset covers an area of 16,441  $\text{km}^2$ , mostly including urban areas, and industrial areas with dense traffic networks; hence, high air quality pollutant indices are usually recorded. It includes hourly recordings of 6 types of pollutants, namely PM<sub>2.5</sub>, PM<sub>10</sub>, NO<sub>2</sub>, CO, SO<sub>2</sub>, and O<sub>3</sub>, over a period of 8643 h. Besides, meteorology features including temperature, evaporation, precipitation, and wind speed are also recorded.

Table 3 summarizes the dataset statistical information, e.g., mean, median, standard deviation (std), and max, min value of the processed datasets.

### 6.1.1 Experimental setup

In accordance with the restriction outlined in [30], our analysis is restricted to stations with adjacent stations located within a distance of 200 km. As a result, the number of stations meeting this criterion is reduced to 30 for the UK dataset, while the Beijing dataset retains 35 stations. These stations are then randomly partitioned into training, validation, and testing sets. During the training phase, the encoder model is trained to learn embeddings from all features, including the target feature (e.g.,  $PM_{2.5}$ ) from all training stations. The decoder then utilizes current information from all training stations to predict the target feature value for the validation stations. In the testing phase, the model's performance is evaluated by predicting the current target feature for the testing stations. The training/validation/testing ratio is set to 7:5:4 for the Beijing and 9:4:4 for the UK datasets, respectively. Figures 5 and 6 illustrate the distribution of monitoring stations within the training, validation, and testing datasets for both the Beijing and UK datasets.

## 6.2 Setting

**Evaluation Indicators** In this work, we use four statistical indicators to evaluate the performance of models, including root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), median absolute percentage error (MdAPE), and  $R^2$  Score ( $R^2$ ). Formulas of these indicators are presented in 16.

$$\begin{aligned}
 RMSE &= \sqrt{\frac{1}{N} \sum_{t=1}^N (y_t - \tilde{y}_t)^2}, \\
 MAE &= \frac{1}{N} \sum_{t=1}^N |y_t - \tilde{y}_t|, \\
 MdAPE &= \text{median} \left( \left| \frac{y_t - \tilde{y}_t}{\tilde{y}_t} \right| \right), \\
 MAPE &= \frac{1}{N} \sum_{t=1}^N \left| \frac{y_t - \tilde{y}_t}{\tilde{y}_t} \right|, \\
 R^2 &= 1 - \frac{\sum_{t=1}^N (\tilde{y}_t - y_t)^2}{\sum_{t=1}^N (y_t - \bar{y}_t)^2},
 \end{aligned} \tag{16}$$

where  $y_t$  and  $\tilde{y}_t$  are the ground truth and the predicted result of the model, respectively. We also use  $\bar{y}_t$ , and  $\bar{\tilde{y}}_t$  standing for the mean values of  $y_t$  and  $\tilde{y}_t$ , respectively.

### 6.2.1 Benchmarks

In order to verify the performance of our proposed model, we compare our method with the deep learning-based techniques for fair comparison, namely KIDW-TCGRU, BiLSTM-IDW, AttPolling FCNN, and FCNN.

- **BiLSTM-IDW**: introduced by Ma et al. [31] in 2019. It is a two-phased model using BiLSTM to learn the output feature embedding and Invert Distance Weight (IDW) to aggregate the feature embedding using a distance-based linear function. The aggregated output feature is then forwarded to a prediction layer to output the final interpolation value.
- **KIDW-TCGRU**: proposed by Guo et al. [32] in 2020. This approach is a combination of the inverse-distance weighting KNN (IDW-KNN) and the TCGRU model. The IDW-KNN method aims to select the nearest stations to perform interpolation. The TCGRU model, which is the combination of time-distributed convolutional neural network (TCNN) and gated recurrent network (GRU), otherwise, helps the model learn spatial and temporal characteristics.
- **AttPolling FCNN**: proposed by Colchado et al. [55]. This method introduces a deep learning model based on an attention mechanism that learns the impact score of neighbor nodes regardless of distance information. Then, the prediction layer consisting of multiple fully connected layers combines the weighted feature vector from neighboring nodes to calculate the  $PM_{2.5}$  concentration value at the target point.
- **FCNN**: This method is a simplified version of the Attention-Polling FCNN method. However, instead of automatically finding the neighboring nodes, these stations are determined using the distance information. This approach then uses the meteorology information and  $PM_{2.5}$  value from neighbor nodes as input for the prediction layer, which consists of multiple fully connected layers, to output the interpolated value.

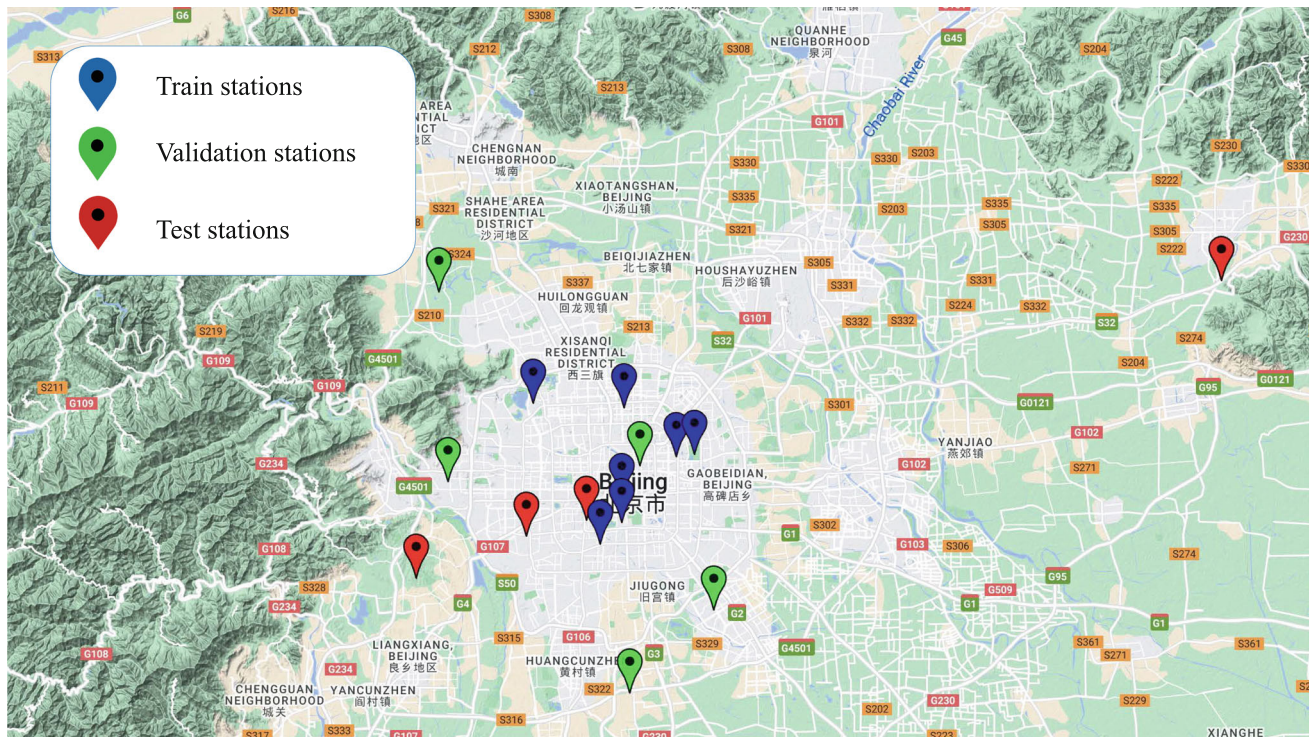
**6.2.1.1 End-to-end comparison** In this section, we evaluate the accuracy of our method (GEDE) in contrast to the baseline methods.

Table 4 shows the detailed performances of proposed methods and baseline models in both datasets. The best experimental results are highlighted in bold for ease of reference. Overall, our proposed model's result exceeds the performance of other baseline models in both datasets.



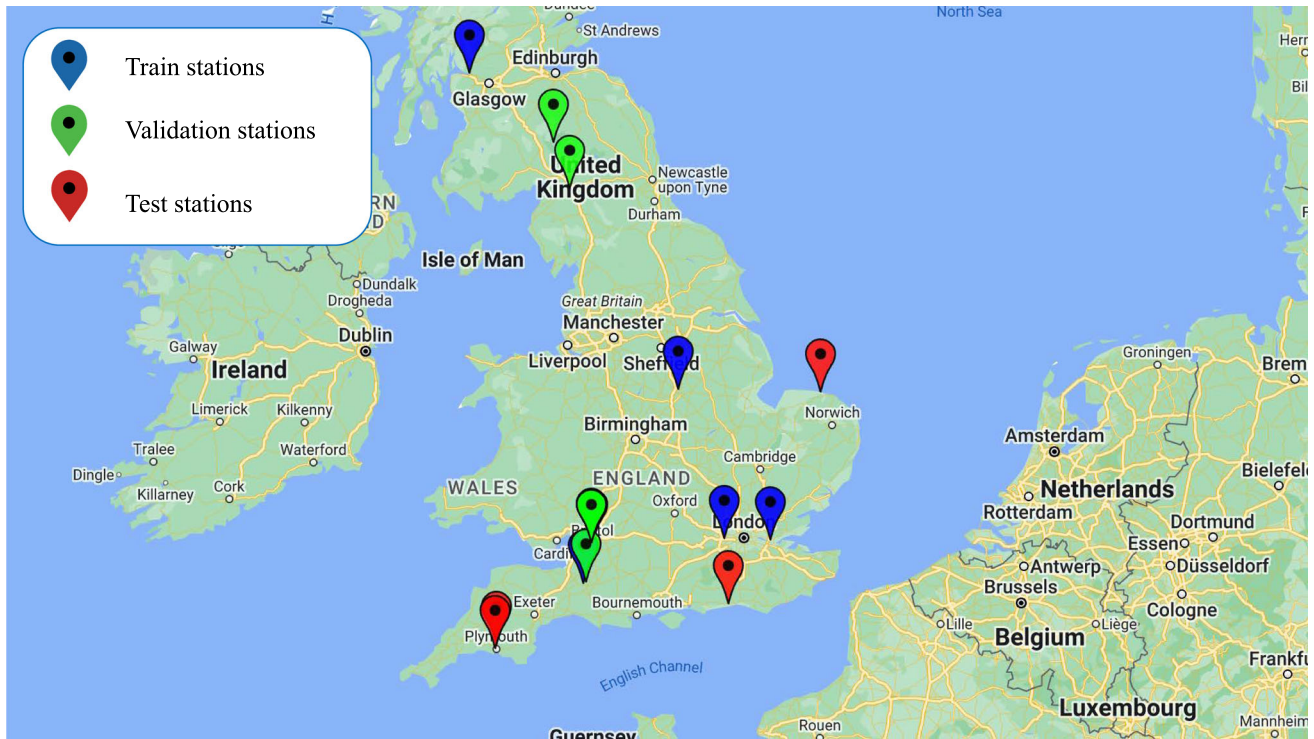
**Table 3** A detailed statistical description of Beijing and UK datasets

Dataset	Feature	Max	Min	Mean	Median	Std
Beijing	CO ( $\mu\text{g}/\text{m}^3$ )	1.9	0.1	0.83	0.7	0.48
	NO <sub>2</sub> ( $\mu\text{g}/\text{m}^3$ )	104.95	3	48.42	43	28.04
	O <sub>3</sub> ( $\mu\text{g}/\text{m}^3$ )	191	1	57.59	46	53.9
	SO <sub>2</sub> ( $\mu\text{g}/\text{m}^3$ )	18	1	5.6	3	4.82
	PM <sub>10</sub> ( $\mu\text{g}/\text{m}^3$ )	232	1	90.4	77.11	60.85
	PM <sub>2.5</sub> ( $\mu\text{g}/\text{m}^3$ )	150.93	3.6	55.3	46.6	39.34
	Temperature ( $^{\circ}\text{C}$ )	39	-16.9	12.17	13.14	13.07
	Evaporation ( $rH^{-3}$ )	99.18	47.8	83.49	84.38	7.6
	Precipitation (mm)	18	0	0.195	0	0.97
	Wind speed (m/s)	8.25	0.03	1.93	1.72	1.16
UK	O <sub>3</sub> ( $\mu\text{g}/\text{m}^3$ )	48.5	0.56	28.24	29.02	7.93
	NO <sub>2</sub> ( $\mu\text{g}/\text{m}^3$ )	23.97	2.31	7.51	6.5	3.28
	SO <sub>2</sub> ( $\mu\text{g}/\text{m}^3$ )	3.76	0.04	0.43	0.36	0.29
	NO <sub>x</sub> ( $\mu\text{g}/\text{m}^3$ )	75.25	3.29	9.68	8.25	5.3
	PM <sub>2.5</sub> ( $\mu\text{g}/\text{m}^3$ )	71.85	2.93	10.76	8.04	7.58
	PM <sub>10</sub> ( $\mu\text{g}/\text{m}^3$ )	92.49	5.45	22.88	21.47	10.26
	Temperature ( $^{\circ}\text{C}$ )	25.74	-7.79	7.19	7.53	4.95
	Evaporation ( $rH^{-3}$ )	170	10.65	39.4	36	24.8
	Precipitation (mm)	45	0	0.3	0.06	0.57
	Wind speed (m/s)	12.2	0.09	4.52	4.36	2.11

**Fig. 5** An illustration of the distribution of the Beijing dataset

Specifically, in terms of MAE error, our proposed model improves from 4.93 to 34.88% in the UK dataset, while this number for the Beijing dataset ranges from 6.89 to 31.94%.

For other indicators, the proposed method's performance still exceeds others'. For example, the average improvement of our proposed model in terms of RMSE is 13.33%



**Fig. 6** An illustration of the distribution of the UK dataset

and 14.37% on the Beijing and UK datasets, respectively. In MAPE indicator, these statistics are 36.05% and 13.25%, while for MDAPE they are 24.48% and 36.33%. Finally, considering the  $R_2$  error, the average improvement of our proposed model in two proposed datasets is 5.39% and 32.58%, respectively.

To further facilitate understanding, we visualize the predicted and ground-truth values in Figs. 7 and 8. Specifically, we choose two specific examples from the two datasets. The first example was obtained from a station named “New North Zone” from 14/12/2018 to 23/12/2018, while the second example was obtained from a station named “EDNS” from 10/06/2019 to 07/11/2019. In both figures, it is noticeable that the proposed method predicts low points significantly better than others. However, due to the nature of the method, which heavily relies on neighbor stations’  $PM_{2.5}$  values, the predicted  $PM_{2.5}$  indicator’s tendency is hugely affected by the tendencies of other neighbor stations. An example for this problem of this proposed method problem is the surge in the predicted data value on the Beijing dataset between the 720th timestep and the 750th timestep.

**6.2.1.2 Ablation study** Aiming to answer the question **RQ2**, we conduct experiments in which we remove each component from the model and record the variation in the performance of models. The detail of each case is detailed as follows:

- **GEDE-1:** We remove the local attention mechanism to examine the impact of this mechanism on the final output result.
- **GEDE-2:** The global attention mechanism is removed from the architecture to explore the importance of this mechanism.
- **GEDE-3:** The graph neural network is removed from the architecture of the encoder. The encoder is then used to train the embedding for the nodes using fully connected layers and the recurrent neural network.
- **GEDE-4:** The recurrent neural network is removed from the architecture of the encoder. Specifically, we want to monitor the recurrent neural network’s impact on the encoder’s learning.
- **GEDE-5:** In this variant, we measure the impact of the meteorology feature embedding layer by removing it from the original architecture. In this specific experiment, the decoder, in this case, estimates the output value using only the air quality features.
- **GEDE-6:** We remove the node-feature corruption from the corrupt function to examine the impact of this approach in learning a highly representative feature embedding.

Table 5 illustrates the results of the mentioned ablation models for several evaluation indicators. We notice that our full model **GEDE** outperforms the other variants, which shows our design choices’ positive impact. In

particular, the final **GEDE** model's performance surpasses these other two attentive variants, e.g., **GEDE-1** and **GEDE-2** at 3.52% and 4.86% in MAE metric, averaged over both datasets. This illustrates the effectiveness of combining both attention mechanisms compared to using only one. A similar drop of averaged MAE at 17.43% and 18.78% can be seen for **GEDE-3** and **GEDE-4**, respectively, which highlights the benefits of using graph neural network and recurrent neural network to learn the spatiotemporal characteristics. However, as the statistics provided, it is noticeable that spatial characteristic provides more impact than temporal characteristic. The full model outperforms the variants **GEDE-5** at 12.5% in MAE metric, which shows the positive impact of the meteorology feature embedding and the node-feature corrupt function. Last but not least, the variant **GEDE-6** has averaged MAE reduction of 16.47% compared to the full model. This indicates the robustness of our node feature corrupt function to the previous approaches, which allows the final model to learn the general feature embedding and adapt well to unseen data.

**6.2.1.3 Feature importance analysis** To address research question RQ3, we conduct two analyses:

1. First, we measure the correlation coefficient of each input feature (excluding PM-related indicators) to the target feature (i.e.,  $PM_{2.5}$ ).
2. Second, we select five features that correlate most to  $PM_{2.5}$  and measure their Shapley values in predicting  $PM_{2.5}$  using our proposed model.

## 6.2.2 Correlation coefficient

The coefficient of each input feature to  $PM_{2.5}$  is plotted in Fig. 9. For the Beijing dataset, the most correlated features are CO, NO<sub>2</sub>, SO<sub>2</sub>, Surface pressure, and Temperature. For the UK dataset, they are NO<sub>2</sub>, SO<sub>2</sub>, O<sub>3</sub>, Surface pressure,

and Temperature. These findings align with existing works on the relatedness to  $PM_{2.5}$  [56–58]. Furthermore, as Fig. 9 illustrates, there is a clear and robust correlation between CO, NO<sub>2</sub>, O<sub>3</sub>, SO<sub>2</sub>, and  $PM_{2.5}$  index, which are also confirmed in [59] and [60]. This strong correlation is likely due to their common sources, stemming from vehicle emissions, industrial processes, and fossil fuel combustion. This connection is particularly evident in densely populated, industrialized, and heavily trafficked areas like Beijing. On the other hand, the temperature feature shows a weaker correlation in both datasets; however, it still positively influences the model's predictive ability, as stated in [58]. It is worth noting that, apart from the air quality index, other indices do not strongly correlate with  $PM_{2.5}$ . Overall, this experiment provides valuable insights into the correlation between  $PM_{2.5}$ , air quality indices, and weather features in the UK and Beijing datasets.

However, the correlation coefficients can only answer the question of which features are most correlated with  $PM_{2.5}$  but not which features are most influential in estimating  $PM_{2.5}$ . To address this question, we adopt the Shapley value [61] to evaluate the influence of each individual input feature on the performance of our proposed model.

## 6.2.3 Shapley values

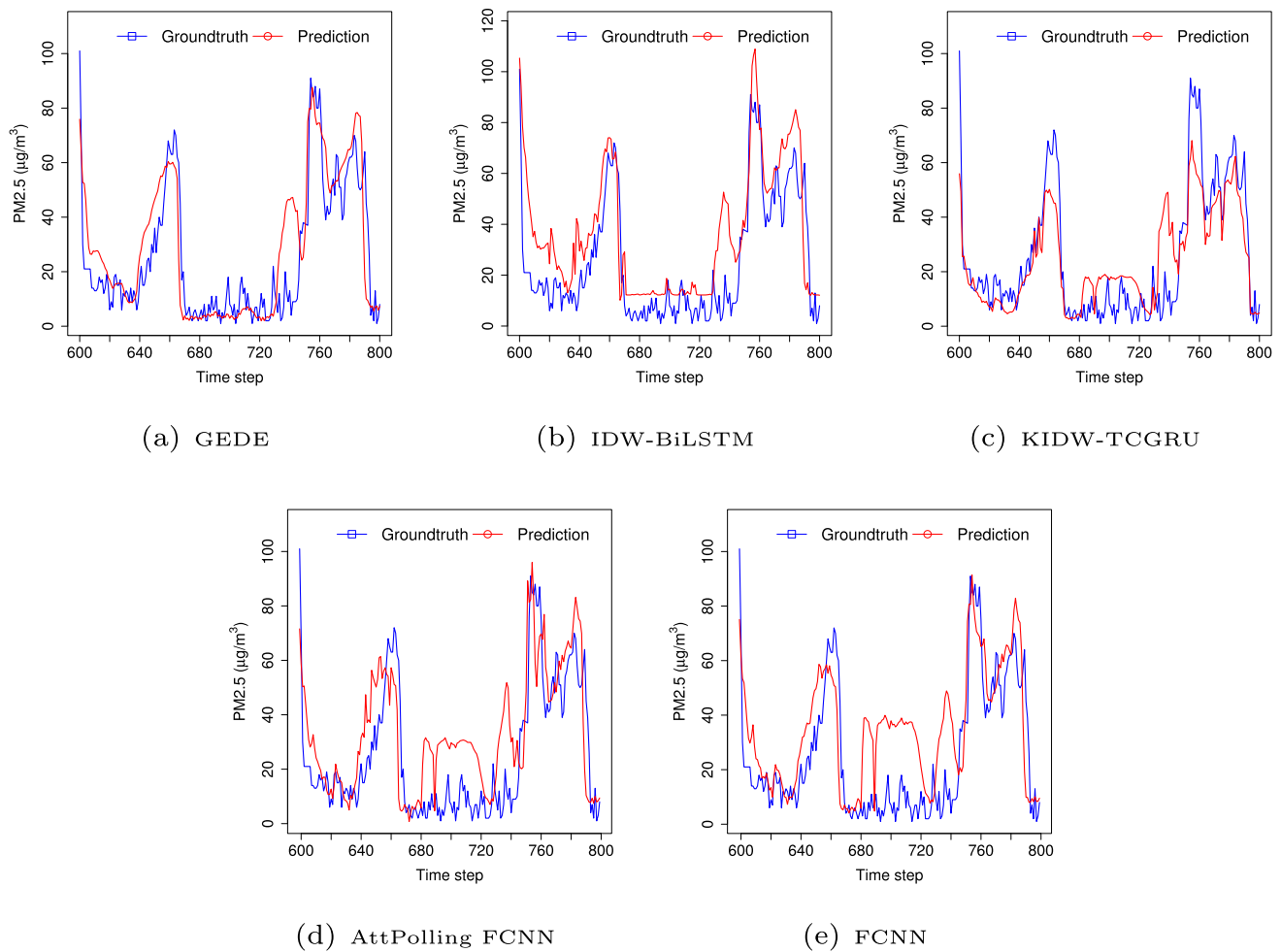
The Shapley value associated with each feature represents the average marginal impact of that feature's value across all possible feature combinations. Let us denote by  $F_k$  an input feature, and  $\mathbf{Z} = \{F_1, \dots, F_m\}$  the set of all input features (excluding PM-related indicators). Moreover, for each subset  $\mathbf{S} \subseteq \mathbf{Z}$ , let us denote by  $val(\mathbf{S})$  the MAE of our proposed model when using  $\mathbf{S}$  and  $PM_{2.5}$  from the monitoring stations to predict  $PM_{2.5}$  at the targeted location. The Shapley value  $\phi_{F_k}(val)$  of a feature  $F_k$  ( $k = 1, \dots, m$ ) is calculated using the following formula:

**Table 4** A detailed comparison of the average accuracy of fine-grained air quality estimation methods

Dataset	Model	MAE	RMSE	MAPE	MdAPE	R2
Beijing	FCNN	11.7	15.36	0.67	0.26	0.86
	BiLSTM-IDW	13.25	18.28	0.61	0.322	0.85
	KIDW-TCGRU	16.28	20.38	0.78	0.432	0.76
	AttPolling FCNN	11.15	15.03	0.63	0.252	0.87
	<b>GEDE</b>	<b>10.6</b>	<b>14.12</b>	<b>0.43</b>	<b>0.239</b>	<b>0.88</b>
UK	FCNN	2.33	3.58	0.38	0.297	0.45
	BiLSTM-IDW	2.59	3.6	0.39	0.39	0.42
	KIDW-TCGRU	2.85	4.18	0.52	0.52	0.43
	AttPolling FCNN	2.32	3.35	0.37	0.257	0.48
	<b>GEDE</b>	<b>2.16</b>	<b>3.19</b>	<b>0.36</b>	<b>0.233</b>	<b>0.59</b>

The best results are highlighted in Bold





**Fig. 7** Visualization of prediction result against the ground-truth on the Beijing dataset

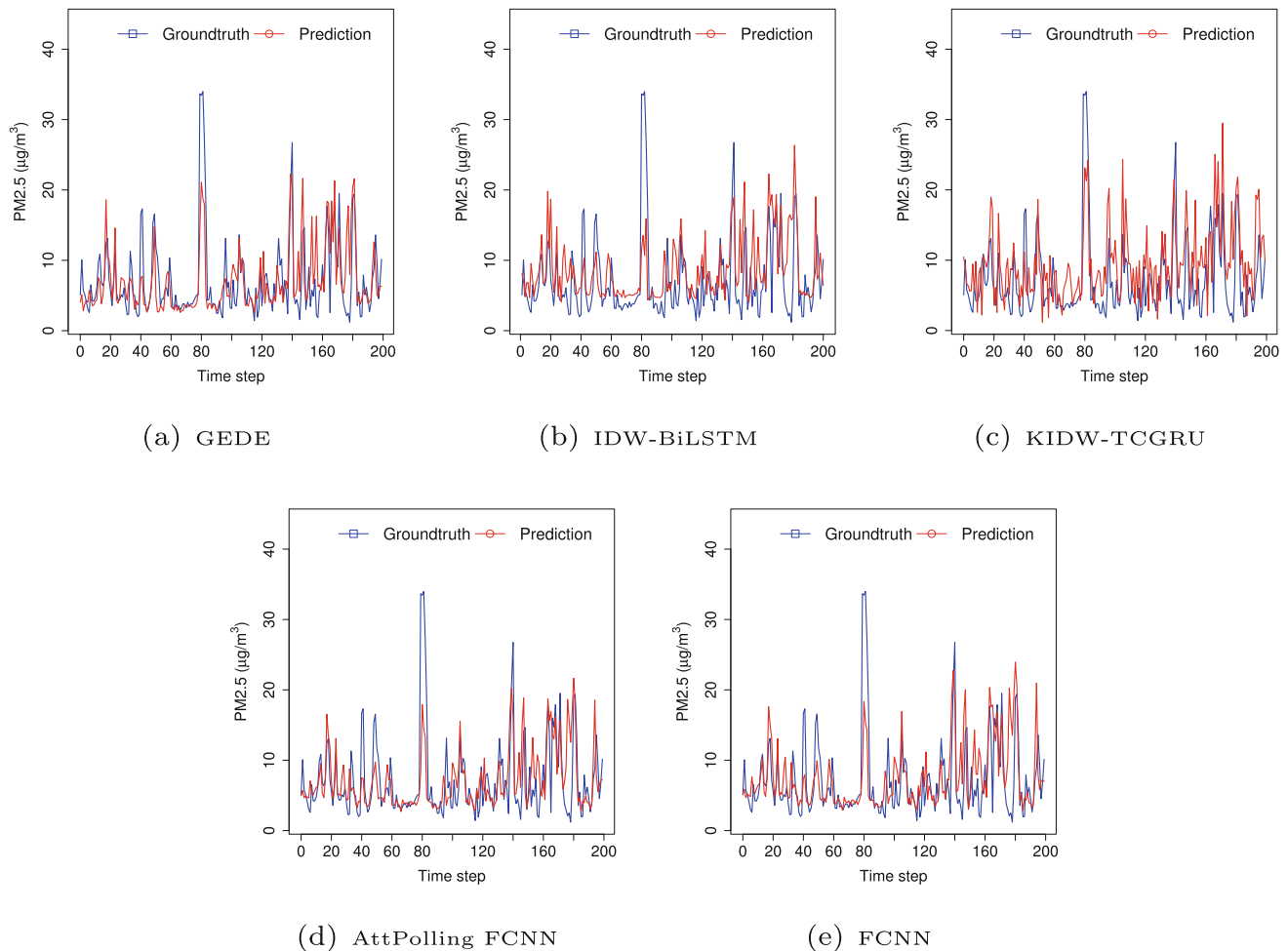
$$\phi_{F_k}(val) = \sum_{S \subseteq \mathbf{Z} \setminus \{F_k\}} \frac{|\mathbf{S}|!(m - |\mathbf{S}| - 1)!}{m!} \quad (17)$$

$$(val(\mathbf{S} \cup \{F_k, PM_{2.5}\}) - val(\mathbf{S} \cup PM_{2.5})).$$

It is worth noting that our model utilizes  $PM_{2.5}$  and other indicators from monitoring stations to predict  $PM_{2.5}$  at an arbitrary location of interest; thus, values of  $PM_{2.5}$  from monitoring stations must be included in the input data. A lower Shapley value indicates a higher influence of the corresponding feature on the estimation accuracy of the  $PM_{2.5}$  indicator. As the number of subset  $\mathbf{S}$  increases exponentially to the cardinality of  $\mathbf{Z}$ , in this experiment, we only investigate the Shapley values of five input features that most correlate to  $PM_{2.5}$ . Specifically, for the Beijing dataset, the selected features encompass  $CO$ ,  $NO_2$ ,  $SO_2$ , Surface pressure, and Temperature. In contrast, the UK dataset's chosen features consist of  $O_3$ ,  $SO_2$ ,  $NO_2$ , Surface Pressure, and Temperature. As illustrated in Fig. 10, the most influential factors impacting the performance of the estimation model in the Beijing dataset are  $NO_2$  and  $SO_2$ ,

ranking first and second, respectively, in terms of influence. In the UK dataset,  $NO_2$  holds the second most significant effect, while  $O_3$  exerts the greatest influence in the UK dataset, which is noteworthy given its close association with the  $PM_{2.5}$  index, as mentioned in [62]. Moreover,  $SO_2$  and  $CO$  also make positive contributions to the model's predictive capabilities, consistent with their relationship with  $PM_{2.5}$  as indicated in [63], where it is stated that “ $PM_{2.5}$  was positively correlated with  $CO$  at both daily and monthly scales.” Given that these features are all linked to air quality characteristics, their positive impact on predicting the  $PM_{2.5}$  index is unsurprising. Furthermore, both temperature datasets exhibit favorable effects; however, their contribution to the model's estimation performance is not on par with the air quality index features.

**6.2.3.1 Impacts of geographical distance on estimation accuracy** In this section, we investigate the impacts of geographical distance on estimation accuracy. Specifically, we fix the set of monitoring stations used in the training phase, which we refer to as training stations. We then vary



**Fig. 8** Visualization of prediction result against the ground-truth on the UK dataset

**Table 5** Effects of different components on model's performance

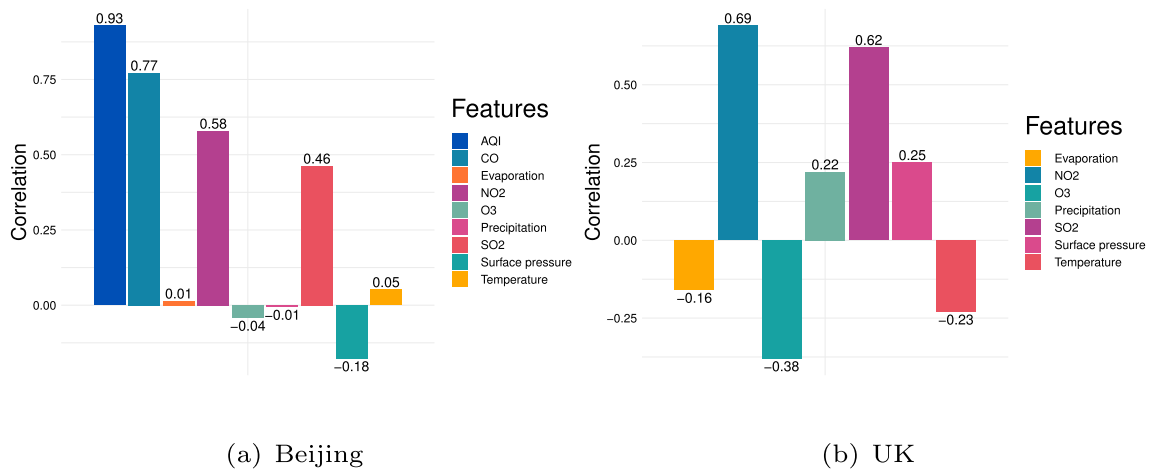
Dataset	Model	GEDE	GEDE-1	GEDE-2	GEDE-3	GEDE-4	GEDE-5	GEDE-6
<b>Beijing</b>	MAE	<b>10.21</b>	10.44	10.93	12.91	13.37	12.23	13.46
	RMSE	<b>15.22</b>	16.06	17.2	17.89	18.89	17.6	25.54
	MdAPE	<b>0.21</b>	0.21	0.25	0.31	0.31	0.28	0.33
	MAPE	<b>0.25</b>	0.48	0.52	0.71	0.7	0.6	0.83
	R2	<b>0.89</b>	0.88	0.86	0.85	0.83	0.85	0.85
<b>UK</b>	MAE	<b>2.16</b>	2.27	2.23	2.51	2.31	2.36	2.35
	RMSE	<b>3.19</b>	3.37	3.2	3.65	3.39	3.52	4.96
	MdAPE	<b>0.23</b>	0.24	0.24	0.28	0.26	0.24	0.24
	MAPE	<b>0.26</b>	0.39	0.39	0.48	0.37	0.39	0.39
	R2	<b>0.39</b>	0.34	0.31	0.18	0.29	0.22	0.21

The best results are highlighted in Bold

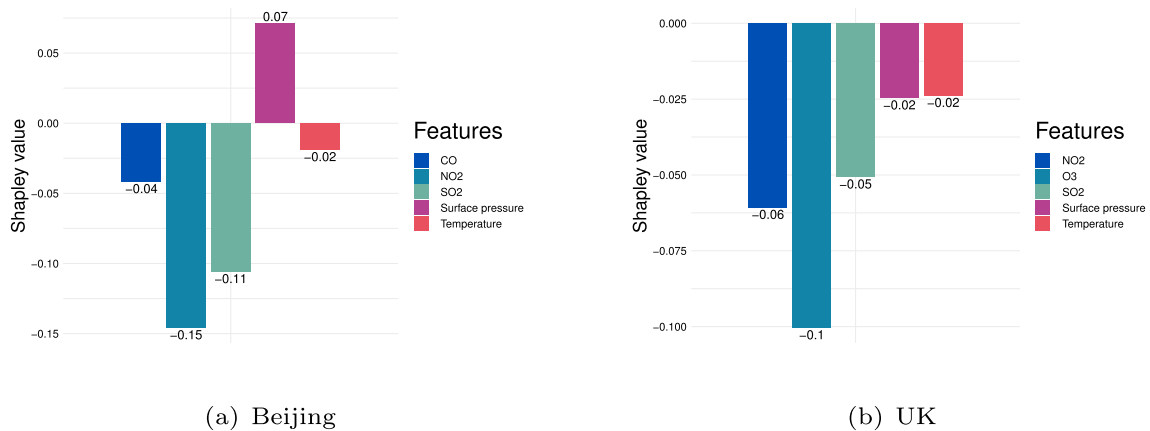
the locations of interest used in the testing phase. (We name these locations as testing locations.) Our objective is to investigate the potential impact of the distance between the testing locations and the training stations on prediction accuracy. The experiment is conducted using the Beijing

dataset. Specifically, we randomly choose 7 training stations in the central region, and 12 testing stations spread throughout the network. The testing stations are categorized into three clusters based on their average distance to the training stations, namely Cluster 1, Cluster 2, and





**Fig. 9** Correlation between  $PM_{2.5}$  with other indicators



**Fig. 10** Shapley value on Beijing and UK datasets

**Cluster 3.** We illustrate the locations of each target station in Fig. 11. The reason for choosing many testing locations (instead of only two) is to guarantee the generalizability of the results.

The results, shown in Table 6, depict that the estimation accuracy is inversely proportional to the distance between testing locations and training stations. Cluster 1, with the shortest average distance of 3.83 kms, achieves a significantly better performance with an MAE indicator of 7.12, 62.07%, and 143.96% higher than Cluster 2 and Cluster 3, respectively.

## 7 Conclusion

This paper presents a novel framework for fine-grained air quality estimation that leverages graph self-supervised representational learning to effectively capture the spatial and temporal dynamics. Specifically, we leverage the

T-GCN model and the contrastive learning paradigm to embed the spatial and temporal characteristics of the input graph before applying a deep-learning-based interpolation method to estimate the target indicator at an arbitrary target location. Furthermore, two attention mechanisms are introduced: Location-aware attention and Feature-aware attention, which capture interstation relationships and emphasize the most significant stations to the location of interest, respectively. The proposed model achieves state-of-the-art performance on the prediction task compared to other baselines. Despite its state-of-the-art performance in prediction tasks compared to other baselines, our proposed model does exhibit certain limitations, such as extended training time and large model size. These are primarily due to the requirement of two training steps of the spatiotemporal graph encoder and the multi-level attention interpolator. In future work, we aim to refine our current methodology to mitigate these weaknesses and enhance the model's estimation accuracy. We hope our work will

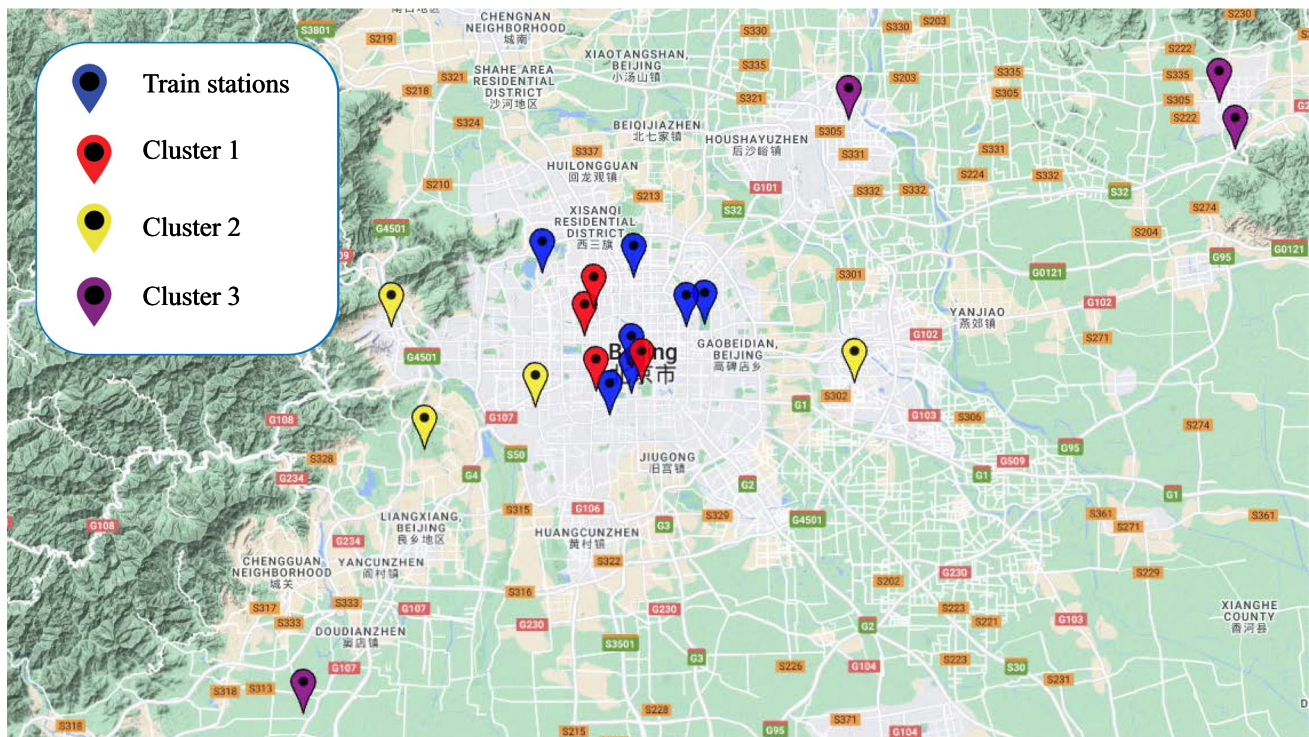


Fig. 11 Visualization of the distribution of train and target stations in the experiment by clusters.

**Table 6** Mean MAE of stations with different distances

Cluster	Mean distance(km)	Mean MAE
1	3.83	7.12
2	14.97	11.54
3	46.01	17.37

encourage and facilitate future research on the air quality estimation problem.

## Appendix A Details of hyper-parameter settings

All our experiment is conducted on NVIDIA GeForce RTX 2080 Ti graphic card. The Cuda version is 11.4. The deep-learning framework PyTorch version 3.8 is used to implement this approach. In our implementation, we use the default batch size of 32 using the Adam optimizer [52]. The self-supervised training of embedding is carried out for 30 epochs, with the initial learning rate of  $1e^{-3}$ . The number of epochs trained for the supervised models is also 30, with the initial learning rate of  $1e^{-3}$ . We use early stopping to get the best model weight. The value of patience in early stopping is 10 epochs.

**Acknowledgements** This work was funded by Vingroup Joint Stock Company (Vingroup JSC), Vingroup, and supported by the Vingroup Innovation Foundation (VINIF) under project code VINIF.2020.DA09. This research is partially funded by Hanoi University of Science and Technology (HUST) under grant number T2022-PC-049. Viet Hung Vu and Duc Long Nguyen were funded by Vingroup Joint Stock Company and supported by the Domestic Master/PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), Vingroup Big Data Institute (VINBIGDATA), under Grant VINIF.2022.Ths.BK.05 and VINIF.2022.Ths.BK.07, respectively.

**Data availability** The code and datasets generated during and/or analyzed during the current study are available in. <https://github.com/duclong1009/Unsupervised-Air-Quality-Estimation>

## Declarations

**Conflict of interest** All authors declare that they have no conflicts of interest.

## References

1. W. H. O. (WHO): Ambient air pollution: A global assessment of exposure and burden of disease (2016)
2. Tai AP, Mickley LJ, Jacob DJ (2010) Correlations between fine particulate matter (pm<sub>2.5</sub>) and meteorological variables in the united states: implications for the sensitivity of pm<sub>2.5</sub> to climate change. *Atmos Environ* 44(32):3976–3984. <https://doi.org/10.1016/j.atmosenv.2010.06.060>
3. Kulmala M (2018) Build a global Earth observatory. Nature Publishing Group

4. Rahmati Aidinlou H, Nikbakht AM (2022) Fuzzy-based modeling of thermohydraulic aspect of solar air heater roughened with inclined broken roughness. *Neural Comput Appl* 34(3):2393–2412. <https://doi.org/10.1007/s00521-021-06547-w>
5. Liu X, Jayaratne R, Thai P, Kuhn T, Zing I, Christensen B, Lamont R, Dunbabin M, Zhu S, Gao J, Wainwright D, Neale D, Kan R, Kirkwood J, Morawska L (2020) Low-cost sensors as an alternative for long-term air quality monitoring. *Environ Res* 185:109438. <https://doi.org/10.1016/j.envres.2020.109438>
6. deSouza P, Anjomshoa A, Duarte F, Kahn R, Kumar P, Ratti C (2020) Air quality monitoring using mobile low-cost sensors mounted on trash-trucks: methods development and lessons learned. *Sustain Cities Soc* 60:102239. <https://doi.org/10.1016/j.scs.2020.102239>
7. Motlagh NH, Lagerspetz E, Nurmi P, Li X, Varjonen S, Miner- aud J, Siekkinen M, Rebeiro-Hargrave A, Hussein T, Petaja T, Kulmala M, Tarkoma S (2020) Toward massive scale air quality monitoring. *IEEE Commun Mag* 58(2):54–59. <https://doi.org/10.1109/MCOM.001.1900515>
8. Idrees Z, Zheng L (2020) Low cost air pollution monitoring systems: a review of protocols and enabling technologies. *J Ind Inf Integr* 17:100123. <https://doi.org/10.1016/j.jii.2019.100123>
9. Lin Y-C, Lee S-J, Ouyang C-S, Wu C-H (2020) Air quality prediction by neuro-fuzzy modeling approach. *Appl Soft Comput* 86:105898. <https://doi.org/10.1016/j.asoc.2019.105898>
10. Xiao X, Jin Z, Wang S, Xu J, Peng Z, Wang R, Shao W, Hui Y (2022) A dual-path dynamic directed graph convolutional network for air quality prediction. *Sci Total Environ* 827:154298. <https://doi.org/10.1016/j.scitotenv.2022.154298>
11. Wang J, Li J, Wang X, Wang J, Huang M (2021) Air quality prediction using CT-LSTM. *Neural Comput Appl* 33(10):4779–4792. <https://doi.org/10.1007/s00521-020-05535-w>
12. Wang J, Song G (2018) A deep spatial-temporal ensemble model for air quality prediction. *Neurocomputing* 314:198–206. <https://doi.org/10.1016/j.neucom.2018.06.049>
13. Han J, Liu H, Zhu H, Xiong H, Dou D (2021) Joint air quality and weather prediction based on multi-adversarial spatiotemporal networks. *Proceed AAAI Conf Artif Intell* 35:4081–4089. <https://doi.org/10.1609/aaai.v35i5.16529>
14. Chen P-C, Lin Y-T (2022) Exposure assessment of pm2.5 using smart spatial interpolation on regulatory air quality stations with clustering of densely-deployed microsensors. *Environ Pollut* 292:118401. <https://doi.org/10.1016/j.envpol.2021.118401>
15. Beauchamp M, Malherbe L, de Fouquet C, Létinois L, Tognet F (2018) A polynomial approximation of the traffic contributions for kriging-based interpolation of urban air quality model. *Environ Modell Softw* 105:132–152. <https://doi.org/10.1016/j.envsoft.2018.03.033>
16. Li J, Heap AD (2011) A review of comparative studies of spatial interpolation methods in environmental sciences: performance and impact factors. *Eco Inform* 6(3):228–241. <https://doi.org/10.1016/j.ecoinf.2010.12.003>
17. Noi E, Murray AT (2022) Interpolation biases in assessing spatial heterogeneity of outdoor air quality in Moscow, Russia. *Land Use Policy* 112:105783. <https://doi.org/10.1016/j.landusepol.2021.105783>
18. Xu C, Wang J, Hu M, Wang W (2022) A new method for interpolation of missing air quality data at monitor stations. *Environ Int* 169:107538. <https://doi.org/10.1016/j.envint.2022.107538>
19. Alimissis A, Philippopoulos K, Tzanis C, Deligiorgi D (2018) Spatial estimation of urban air pollution with the use of artificial neural network models. *Atmos Environ* 191:205–213. <https://doi.org/10.1016/j.atmosenv.2018.07.058>
20. Ma J, Ding Y, Cheng JC, Jiang F, Wan Z (2019) A temporal-spatial interpolation and extrapolation method based on geographic long short-term memory neural network for pm 2.5. *J Clean Prod* 237:117729. <https://doi.org/10.1016/j.jclepro.2019.117729>
21. Qi Z, Wang T, Song G, Hu W, Li X, Zhang Z (2018) Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality. *IEEE Trans Knowl Data Eng* 30(12):2285–2297. <https://doi.org/10.1109/TKDE.2018.2823740>
22. Li L, Girguis M, Lurmann F, Pavlovic N, McClure C, Franklin M, Wu J, Oman LD, Breton C, Gilliland F, Habre R (2020) Ensemble-based deep learning for estimating pm2.5 over California with multisource big data including wildfire smoke. *Environ Int* 145:106143. <https://doi.org/10.1016/j.envint.2020.106143>
23. Rijal N, Gutta RT, Cao T, Lin J, Bo Q, Zhang J (2018) Ensemble of deep neural networks for estimating particulate matter from images. In: 2018 IEEE 3rd International conference on image, vision and computing (ICIVC), pp 733–738. <https://doi.org/10.1109/ICIVC.2018.8492790>
24. Dixit E, Jindal V (2022) Ieeseep: an intelligent energy efficient stable election routing protocol in air pollution monitoring WSNS. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-022-07027-5>
25. Ari D, Alagoz BB (2022) An effective integrated genetic programming and neural network model for electronic nose calibration of air pollution monitoring application. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-022-07129-0>
26. Al-Janabi S, Alkaim A, Al-Janabi E, Aljeboree A, Mustafa M (2021) Intelligent forecaster of concentrations (pm2. 5, pm10, no2, co, o3, so2) caused air pollution (IFCSAP). *Neural Comput Appl* 33(21):14199–14229. <https://doi.org/10.1007/s00521-021-06067-7>
27. Wardana I, Gardner JW, Fahmy SA (2022) Estimation of missing air pollutant data using a spatiotemporal convolutional autoencoder. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-022-07224-2>
28. Liang Y, Ke S, Zhang J, Yi X, Zheng Y (2018) Geoman: Multi-level attention networks for geo-sensory time series prediction. In: Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI-18, pp 3428–3434. <https://doi.org/10.24963/ijcai.2018/476>
29. Zhao J, Deng F, Cai Y, Chen J (2018) Long short-term memory–fully connected (LSTM-FC) neural network for pm2.5 concentration prediction. *Chemosphere*. <https://doi.org/10.1016/j.chemosphere.2018.12.128>
30. Qi Y, Li Q, Karimian H, Liu D (2019) A hybrid model for spatiotemporal forecasting of pm2.5 based on graph convolutional neural network and long short-term memory. *Sci Total Environ*. <https://doi.org/10.1016/j.scitotenv.2019.01.333>
31. Ma J, Ding Y, Gan VJL, Lin C, Wan Z (2019) Spatiotemporal prediction of pm2.5 concentrations at different time granularities using IDW-BLSTM. *IEEE Access* 7:107897–107907
32. Guo C, Liu G, Lyu L, Chen CH (2020) An unsupervised pm2.5 estimation method with different Spatio-temporal resolutions based on KIDW-TCGRU. *IEEE Access* 8:190263–190276. <https://doi.org/10.1109/ACCESS.2020.3032420>
33. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International conference on learning representations. ICLR '17. <https://doi.org/10.48550/ARXIV.1609.02907>
34. Liu Y, Jin M, Pan S, Zhou C, Zheng Y, Xia F, Yu P (2022) Graph self-supervised learning: a survey. *IEEE Transactions on knowledge and data engineering* abs/2103.00111, 1–1. <https://doi.org/10.1109/TKDE.2022.3172903>
35. Kipf TN, Welling M (2016) Variational graph auto-encoders. *CoRR* abs/1611.07308. 1611.07308. <https://doi.org/10.48550/ARXIV.1611.07308>




36. Wang C, Pan S, Long G, Zhu X, Jiang J (2017) Mgae: marginalized graph autoencoder for graph clustering. In: Proceedings of the 2017 ACM on conference on information and knowledge management. CIKM '17, pp. 889–898. <https://doi.org/10.1145/3132847.3132967>
37. Jin W, Derr T, Liu H, Wang Y, Wang S, Liu Z, Tang J (2020) Self-supervised learning on graphs: deep insights and new direction. CoRR **abs/2006.10141**. <https://doi.org/10.48550/ARXIV.2006.10141>
38. Hu Z, Fan C, Chen T, Chang K-W, Sun Y (2019) Pre-training graph neural networks for generic structural feature extraction. In: ICLR 2019 Workshop: representation learning on graphs and manifolds. <https://doi.org/10.48550/ARXIV.1905.13728>
39. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International conference on knowledge discovery and data mining. KDD '14, pp 701–710. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/2623330.2623732>
40. Grover A, Leskovec J (2016) Node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International conference on knowledge discovery and data mining. KDD '16, pp 855–864. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/2939672.2939754>
41. Zhu Y, Xu Y, Yu F, Liu Q, Wu S, Wang L (2020) Deep Graph Contrastive Representation Learning. In: ICML Workshop on Graph Representation Learning and Beyond. <https://doi.org/10.48550/ARXIV.2006.04131>
42. Hamilton WL, Ying R, Leskovec J (2017) Inductive representation learning on large graphs. NIPS'17, pp 1025–1035. Curran Associates Inc., Red Hook, NY, USA. <https://doi.org/10.48550/ARXIV.1706.02216>
43. Velickovic P, Fedus W, Hamilton WL, Liò P, Bengio Y, Hjelm RD (2019) Deep graph infomax. In: 7th International conference on learning representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019. <https://doi.org/10.48550/ARXIV.1809.10341>
44. Opolka FL, Solomon A, Cangea C, Velickovic P, Liò P, Hjelm RD (2019) Spatio-temporal deep graph infomax. ICLR 2019 **abs/1904.06316**. <https://doi.org/10.48550/ARXIV.1904.06316>
45. Winarno E, Hadikurniawati W, Rosso RN (2017) Location based service for presence system using haversine method. In: 2017 International conference on innovative and creative information technology (ICITech), pp 1–4. <https://doi.org/10.1109/INNOCIT.2017.8319153>. IEEE
46. copernicus: ERA5 Hourly Data on Single Levels from 1959 to Present. <https://doi.org/10.24381/cds.adbb2d47>. <https://cds.climate.copernicus.eu/cdsapp/#/dataset/reanalysis-era5-single-levels> Accessed 2019-09-30
47. Li S, Xie G, Ren J, Guo L, Yang Y, Xu X (2020) Urban pm2.5 concentration prediction via attention-based CNN-LSTM. Appl Ci. <https://doi.org/10.3390/app10061953>
48. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. In: NIPS 2014 Workshop on deep learning, December. <https://doi.org/10.48550/ARXIV.1412.3555>
49. Tobler WR (1970) A computer movie simulating urban growth in the Detroit region. Econ Geogr 46:234–240. <https://doi.org/10.2307/143141>
50. Cichowicz R, Wielgosinski G, Fetter W (2020) Effect of wind speed on the level of particulate matter pm10 concentration in atmospheric air during winter season in vicinity of large combustion plant. J Atmos Chem 77:1–14. <https://doi.org/10.1007/s10874-020-09401-w>
51. Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, Deng M, Li H (2020) T-GCN: a temporal graph convolutional network for traffic prediction. IEEE Trans Intell Transp Syst 21(9):3848–3858. <https://doi.org/10.1109/tits.2019.2935152>
52. Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings. <https://doi.org/10.48550/ARXIV.1412.6980>
53. Reani M, Lowe D, Gledson A, Topping D, Jay C (2022) UK daily meteorology, air quality, and pollen measurements for 2016–2019, with estimates for missing data. Sci Data 9(1):43. <https://doi.org/10.1038/s41597-022-01135-6>
54. Wang H air pollution and meteorological data in Beijing 2017–2018. <https://doi.org/10.7910/DVN/USXCAK>
55. Colchado LE, Villanueva E, Ochoa-Luna J (2021) A neural network architecture with an attention-based layer for spatial prediction of fine particulate matter. In: 2021 IEEE 8th International conference on data science and advanced analytics (DSAA), pp 1–10. <https://doi.org/10.1109/DSAA53316.2021.9564200>
56. Chen Y, Zang L, Du W, Xu D, Shen G, Zhang Q, Zou Q, Chen J, Zhao M, Yao D (2018) Ambient air pollution of particles and gas pollutants, and the predicted health risks from long-term exposure to pm25 in zhejiang province, china. Environ Sci Pollut Res 25(24):23833–23844. <https://doi.org/10.1007/s11356-018-2420-5>
57. Chen Z, Xie X, Cai J, Chen D, Gao B, He B, Cheng N, Xu B (2018) Understanding meteorological influences on pm<sub>2.5</sub> concentrations across china: a temporal and spatial perspective. Atmos Chem Phys 18(8):5343–5358
58. Wang J, Ogawa S (2015) Effects of meteorological conditions on pm2.5 concentrations in Nagasaki, Japan. Int J Environ Res Public Health 12:9089–101. <https://doi.org/10.3390/ijerph120809089>
59. Mi K, Zhuang R, Zhang Z, Gao J, Pei Q (2019) Spatiotemporal characteristics of pm2.5 and its associated gas pollutants, a case in china. Sustain Cities Soc 45:287–295. <https://doi.org/10.1016/j.scs.2018.11.004>
60. Li K, Bai K (2019) International Journal of Environmental Research and Public Health. Spatiotemporal Assoc Between pm2.5 So2 Well No2 China From 2015 to 2018 16(13):2352. <https://doi.org/10.3390/ijerph16132352>
61. Hart S In: Eatwell, J., Milgate, M., Newman, P. (eds.) Shapley Value, pp 210–216. Palgrave Macmillan UK, London (1989). [https://doi.org/10.1007/978-1-349-20181-5\\_25](https://doi.org/10.1007/978-1-349-20181-5_25)
62. Jia M, Zhao T, Cheng X, Gong S, Zhang X, Tang L, Liu D, Wu X, Wang L, Chen Y (2017) Inverse relations of pm2.5 and o3 in air compound pollution between cold and hot seasons over an urban area of east china. Atmosphere. <https://doi.org/10.3390/atmos8030059>
63. Fu H, Zhang Y, Liao C, Mao L, Wang Z, Hong N (2020) Investigating PM(2.5) responses to other air pollutants and meteorological factors across multiple temporal scales. Sci Rep 10(1):15639. <https://doi.org/10.1038/s41598-020-72722-z>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Authors and Affiliations

Viet Hung Vu<sup>1</sup> · Duc Long Nguyen<sup>1</sup> · Thanh Hung Nguyen<sup>1</sup> · Quoc Viet Hung Nguyen<sup>2</sup>  · Phi Le Nguyen<sup>1</sup> · Thanh Trung Huynh<sup>3</sup>

✉ Thanh Hung Nguyen  
hungnt@soict.hust.edu.vn

Viet Hung Vu  
hung.vv221026m@sis.hust.edu.vn

Duc Long Nguyen  
long.nd222179m@sis.hust.edu.vn

Quoc Viet Hung Nguyen  
quocviethung1@gmail.com

Phi Le Nguyen  
lenp@soict.hust.edu.vn

Thanh Trung Huynh  
thanh.huynh@epfl.ch

<sup>1</sup> Hanoi University of Science and Technology, Hanoi, Vietnam

<sup>2</sup> Griffith University, Gold Coast, Queensland, Australia

<sup>3</sup> The École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland