Semi-Supervised Air Quality Forecasting via Self-Supervised Hierarchical Graph Neural Network

Jindong Han, Hao Liu, Haoyi Xiong, and Jing Yang

Abstract—Predicting air quality in fine spatiotemporal granularity is of great importance for air pollution control and urban sustainability. However, existing studies are either focused on predicting station-wise future air quality, or inferring current air quality for unmonitored regions. How to accurately forecast future air quality for these unmonitored regions in a fine granularity remains an unexplored problem. In this paper, we propose the Self-Supervised Hierarchical Graph Neural Network (SSH-GNN), for fine-grained air quality forecasting in a semi-supervised way. Specifically, to augment spatially sparse air quality observations, SSH-GNN first approximates the city-wide air quality distribution based on historical readings and various urban contextual factors (e.g., weather conditions and traffic flows). Then, we propose a hierarchical recurrent graph neural network to make city-wide predictions, which encodes the spatial hierarchy of urban regions for long-range spatiotemporal correlation modeling. Moreover, by leveraging spatiotemporal self-supervision strategies, SSH-GNN exploits both universal topological and contextual patterns to further enhance the forecasting effectiveness. Extensive experiments on two real-world datasets show that SSH-GNN significantly outperforms the state-of-the-art algorithms.

Index Terms—Air quality forecasting, graph neural network, self-supervised learning, urban computing

1 Introduction

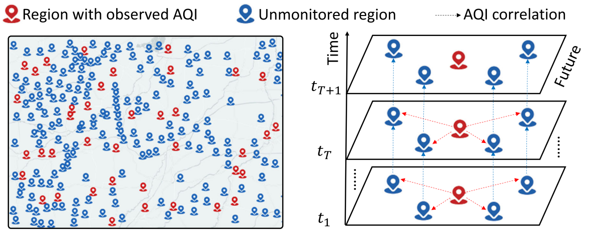
While the rapid progress of urbanization has brought us great convenience in living, working, transportation, etc., it inevitably causes serious air pollution concerns worldwidely. As reported by the World Health Organization (WHO), air pollution has become the world's largest environmental health risk, which is responsible for at least seven million deaths every year [1]. In addition to health issues, air pollution has also created a substantial burden on the economy in many developing countries [2]. In fact, the potential losses caused by air pollution could be reduced or even prevented through effective pollution emission controls and early interventions, if we can accurately predict the air quality index (AQI) in each region of a city ahead of time. Therefore, fine-grained air quality prediction is of great importance to human livelihood, public health, and local economic development.

Prior works of air quality forecasting are either focused on monitoring station-level [3], [4], [5] or regarding the city as a whole [6], [7], but overlook the fine-grained urban areas without monitoring stations deployed. On the other hand, some studies infer the current air quality status of unobserved regions by leveraging observations from monitoring stations and urban data [8], [9], but are incapable of making predictions of the future. As the air quality between different urban regions is influenced by complex urban factors (functionality, urban canyon distribution, and traffic patterns) [8] and may be very different, knowing the future air quality of fine-grained regions is beneficial for various environmental surveillance and governments' policy-making tasks, especially for these regions without monitoring station. In this work, as shown in Figs. 1 and 2, we aim to simultaneously predict air quality for all unmonitored urban regions in the city, based on sparse air quality observations from a few air quality monitoring stations and rich spatiotemporal features derived from multi-sourced urban data (e.g., POI distribution, traffic flows).

However, three non-trivial challenges arise to achieve the goal. (1) *Limited monitoring stations*. For space and economic concern, only a small portion of regions are deployed with air quality monitoring stations. For instance, by partitioning the Beijing city into fine-grained administrative divisions (i.e. streets, townships), only 3.5% streets and 8.8% towns are covered by monitoring stations. That is, there are no real-time air quality readings in the majority of regions without monitoring stations. How to make accurate predictions for these unmonitored regions in a city by leveraging the sparse air quality observations is the first challenge.

- *Jindong Han is with Artificial Intelligence Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong Province 510000, China. E-mail: jhanao@connect.ust.hk.*
- *Hao Liu is with Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, Guangdong Province 510000, China, and also with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR 999077, China. E-mail: liuh@ust.hk.*
- *Haoyi Xiong is with Baidu Research, Beijing 100085, China. E-mail: xionghaoyi@baidu.com.*
- *Jing Yang is with the Environmental Development Center, Ministry of Ecology and Environment, Beijing 100006, China. E-mail: yangjing01@edcmep.org.cn.*

Manuscript received 8 July 2021; revised 20 Dec. 2021; accepted 29 Jan. 2022. Date of publication 9 Feb. 2022; date of current version 3 Apr. 2023.
This work was supported by the National Natural Science Foundation of China under Grant 62102110.
(Corresponding author: Hao Liu.)
Recommended for acceptance by M. A. Cheema.
Digital Object Identifier no. 10.1109/TKDE.2022.3149815



(a) Urban regions

(b) The goal of our work

Fig. 1. Example of semi-supervised air quality forecasting. We aim to forecast AQI for majority unmonitored regions based on a few regions with observed air quality.

(2) *Dynamic long-range spatiotemporal dependencies*. Tobler's first law of geography has been widely adopted to model the air quality variation locality [10]. However, we observe the air quality of distant regions also exhibit spatiotemporal dependencies, which can be utilized to improve the prediction accuracy. On the one hand, two distant regions with similar functionality may have similar air quality patterns, because of the synchronized human activity and urban dynamics [11]. On the other hand, with the effect of atmospheric conditions (e.g., wind, humidity), air pollutants can transport across two distant regions (even different cities) in a very short time [12]. How to model such complex long-range spatiotemporal dependencies for city-wide air quality prediction is another challenge. (3) *Distribution discrepancy between regions*. The air quality variation of different regions may follow a different distribution, which is determined by the diversified environmental context [13]. Learning forecasting models based on observations from regions with monitoring stations may lead to biased results. How to model the distribution shift of unmonitored regions to make generalizable predictions is the third challenge.

To this end, in this paper, we propose the Self-Supervised Hierarchical Graph Neural Network (SSH-GNN), a novel deep learning based framework for semi-supervised fine-grained air quality forecasting. Specifically, based on the sparse real-time air quality observations, we address the first challenge by approximating the missing real-time air quality of unmonitored regions from spatial, temporal, and contextual domains. After that, we propose a hierarchical recurrent graph neural network to capture the dynamic long-range spatiotemporal dependencies. In particular, by constructing a three-level hierarchy, *cities→functional zones→regions*, our hierarchical graph network encodes long-range spatiotemporal dependencies by propagating shared information from top-level city nodes to bottom-level regions. Moreover, we devise tailor-designed spatiotemporal self-supervision strategies to alleviate the distribution shift between regions with and without air quality observations. By exploiting rich topological and contextual information as self-supervision signals, SSH-GNN effectively distills transferable knowledge of unmonitored regions for accurate air quality forecasting.

To our knowledge, this is the first work that exploits the urban hierarchy and self-supervised techniques for semi-supervised fine-grained air quality forecasting. Our research contributions are summarized as follows:

1) We investigate fine-grained air quality forecasting for unmonitored urban regions by formulating the problem as a semi-supervised prediction task.

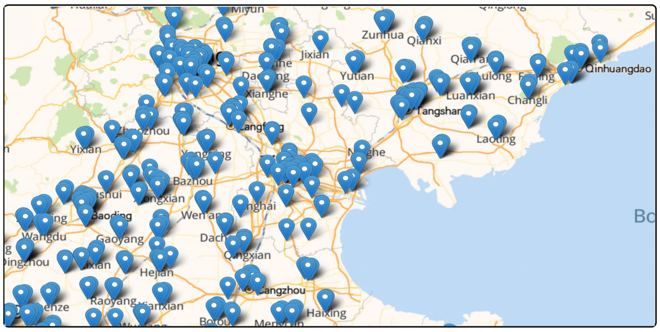


Fig. 2. Spatial distribution of real-world monitoring stations in Beijing-Tianjin-Hebei metropolitan area.

2) We propose a hierarchical spatiotemporal network to capture dynamic long-range spatiotemporal dependencies in an end-to-end way.
3) We introduce spatiotemporal self-supervision techniques for the air quality forecasting task, which further improves the model's generalizability and forecasting effectiveness.
4) We conduct extensive experiments on two real-world datasets collected from the Beijing-Tianjin-Hebei and the Pearl River Delta urban agglomerations, and the results demonstrate the effectiveness of the proposed approach.

The rest of this paper is organized as follows. We first give some preliminaries in Section 2. Then Section 3 is an overview of the proposed framework. After that, we elaborate on model details in Section 4. Moreover, we also describe experimental results to demonstrate the effectiveness of our proposed approach in Section 5. Section 6 presents related works. Finally, we draw conclusion in the last section.

2 Preliminaries

In this section, we first introduce some important definitions and then formalize the semi-supervised air quality forecasting problem we aim to investigate. The notations and explanations we will use throughout the paper are summarized in Table 1.

Definition 1 (Region). *We divide each city into a set of disjoint regions based on the standard township-level administrative division, denoted by $\mathcal{R}$. Each region $r_i \in \mathcal{R}$ is a human settlement associated with a name, a geographical location l_i (i.e., latitude and longitude), and other optional attributes.*

Definition 2 (Functional zone). *A functional zone $z \in \mathcal{Z}$ is made up by multiple regions, serving as a kind of urban functionality, e.g., ecological areas and industrial areas.*

Note the regions in a functional zone may not spatially adjacent with each other, such as multiple business districts in a city.

Definition 3 (City). *A city $c \in \mathcal{C}$ is a set of functional zones providing various functions, e.g., administration, economy, culture and transportation.*

Regions, *functional zones*, and *cities* naturally form a three-level hierarchy from bottom to top, which encoding the city's

TABLE 1
Notations and Explanations

Notation	Explanation
$\mathcal{R}$	region set
$\mathcal{R}_l$	region set with observed air quality
$\mathcal{R}_u$	unmonitored region set
$\mathcal{Z}$	functional zone set
$\mathcal{C}$	city set
$\mathcal{G}^h$	hierarchical region graph
$\mathbf{A}^R, \mathbf{A}^Z, \mathbf{A}^C$	adjacency matrices
$\mathbf{A}^{RZ}, \mathbf{A}^{ZC}$	region-to-zone and zone-to-city assignment matrix
$\mathbf{S}^{RZ}, \mathbf{S}^{ZC}$	soft assignment matrix
$\mathbf{M}^{RZ}, \mathbf{M}^{ZC}$	indication matrix
$\mathbf{x}_i^{a,t}$	air quality of region i at time step t
$\mathbf{x}_i^{m,t}$	meteorological features of region i at time step t
$\mathbf{x}_i^{w,t}$	weather forecast features of region i at time step t
$\mathbf{x}_i^{c,t}$	contextual features of region i at time step t
$\mathbf{x}_i^d, \mathbf{x}_i^e, \mathbf{x}_i^s$	representation of spatial, temporal and contextual views
$\mathbf{x}_i^u$	output of multi-view learning module
$\mathbf{X}^m$	meteorological features of all regions
$\mathbf{X}^c$	contextual features of all regions
N_R, N_Z, N_C	the number of regions, functional zones, and cities
τ	prediction horizon length
$\mathbf{w}, \mathbf{W}$	model parameters
$\sigma(\cdot)$	activation function
$\mathrm{GConv}(\cdot)$	latent factors of user u, item i
λ, β	hyper-parameters controlling the importance of losses

spatial structure and can be exploited to capture region's long-range dependency.

Definition 4 (Hierarchical region graph). *A hierarchical region graph (HRG) is defined as $\mathcal{G}^h = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \mathcal{R} \cup \mathcal{Z} \cup \mathcal{C}$ are vertices including regions, functional zones and cities, and $\mathcal{E} = \{\mathbf{A}^R, \mathbf{A}^Z, \mathbf{A}^C, \mathbf{A}^{RZ}, \mathbf{A}^{ZC}\}$ are edges between each vertex. Specifically, $\mathbf{A}^R$, $\mathbf{A}^Z$ and $\mathbf{A}^C$ are adjacency matrices indicating the connectivity between (1) two region nodes, (2) two zone nodes, (3) two city nodes, $\mathbf{A}^{RZ}$ and $\mathbf{A}^{ZC}$ are region-to-zone and zone-to-city weighted matrices, respectively.*

Note that the regions and cities are real-world administrative divisions, while the functional zones are virtual nodes we aim to learn. The details of hierarchical region graph construction will be introduced in Section 3.3.

Definition 5 (Region air quality). *Given a region $r_i \in \mathcal{R}$ with deployed air-quality monitoring station, the air quality of region r_i at time step t is denoted as $\mathbf{x}_i^{a,t}$, which is a vector of observed air pollutant concentrations, such as AQI, PM2.5 and CO.*

Due to the monitoring station sparsity, only a few regions (less than 12.3% and 8.3% in our datasets) have region air quality. Following processing in [14], if a region has more than one monitoring station, we average observations from different stations as the region air quality.

Consider a set of regions $\mathcal{R} = \mathcal{R}_l \cup \mathcal{R}_u = \{r_1, r_2, \ldots, r_N\}$, where N is total number of regions, $\mathcal{R}_l$ and $\mathcal{R}_u$ are respectively region sets with and without observed region air quality. We use $\mathbf{X}^{a,t} = \{\mathbf{x}_i^{a,t}\}_{i=1}^{|\mathcal{R}_l|}$ to denote all the observed region air quality of R_l at time t. We further associate each region with a set of time-dependent meteorological features $\mathbf{x}_i^{m,t} \in \mathbf{X}^{m,t}$ and weather forecasting features $\mathbf{x}_i^{w,t} \in \mathbf{X}^{w,t}$ at time t, as well as contextual features $\mathbf{x}_i^c \in \mathbf{X}^c$ (e.g., POI distribution, road network, and urban traffic flows).

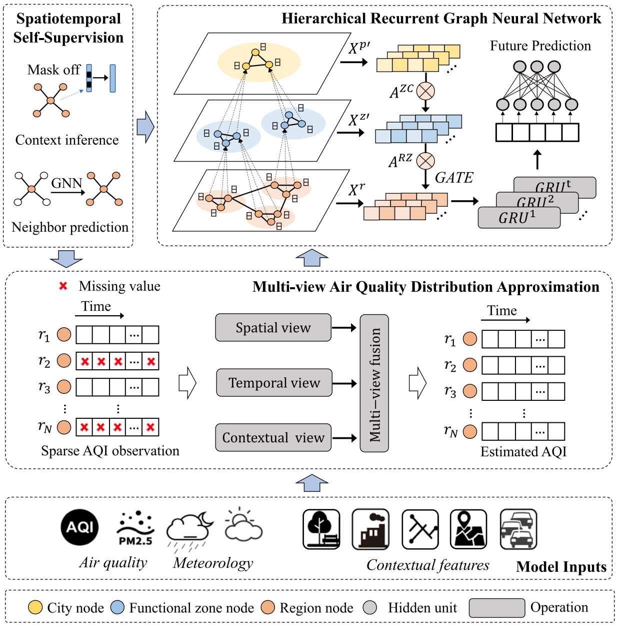


Fig. 3. An overview of SSH-GNN.

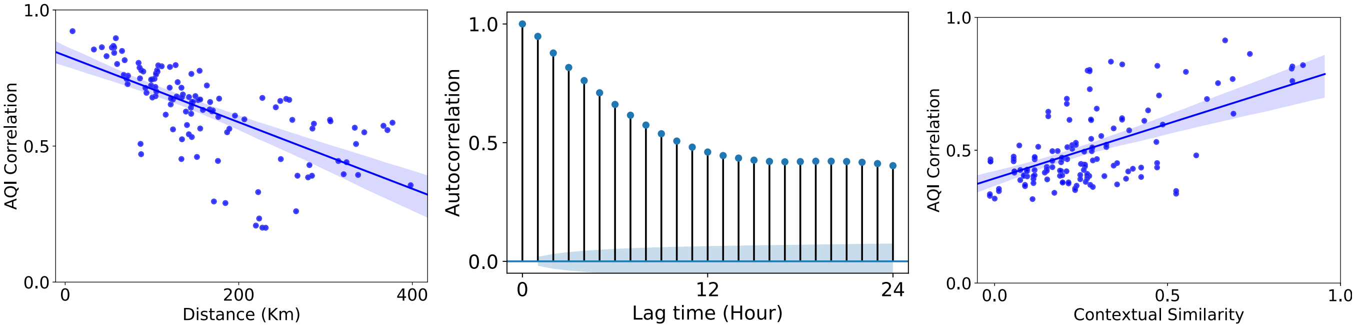
Problem 1. Semi-supervised air quality forecasting. *Given a set of observed historical air quality $\mathcal{X}^a = \{\mathbf{X}^{a,t}\}_{t=1}^T$ for regions $\mathcal{R}_l$, meteorological features $\mathcal{X}^m = \{\mathbf{X}^{m,t}\}_{t=1}^T$, current weather forecasting features $\mathbf{X}^{w,t}$, and contextual features $\mathbf{X}^c$ for all regions $\mathcal{R}$, our goal is to predict region air quality for all $r_i \in \mathcal{R}_u$ over the next τ time steps,*

$$(\hat{\mathbf{y}}^{t+1}, \hat{\mathbf{y}}^{t+2}, \ldots, \hat{\mathbf{y}}^{t+\tau}) \leftarrow \mathcal{F}(\mathcal{X}^a, \mathcal{X}^m, \mathbf{X}^{w,t}, \mathbf{X}^c, \mathcal{G}^h), \tag{1}$$

where $\hat{\mathbf{y}}^{t+1}$ is the predicted air quality at time step $t+1$, and $\mathcal{F}(\cdot)$ is the forecasting function we aim to learn.

3 Framework Overview

The architecture of SSH-GNN is shown in Fig. 3. Based on sparse historical air quality observations and various contextual features as the input, the model outputs future air quality for all unmonitored regions $\mathcal{R}_u$ in the next τ time steps. Overall, there are three major tasks in our approach: (1) approximating the current regional air quality distribution for $\mathcal{R}_u$; (2) learning the dynamic long-range spatiotemporal dependencies via a hierarchical recurrent graph neural network; and (3) exploiting rich unlabeled information through spatiotemporal self-supervision. Specifically, in the first task, we propose a multi-view learning module to jointly infer current region air quality distribution from both spatial, temporal, and contextual domains. In the second task, we propose a hierarchical recurrent graph neural network by constructing a three-level hierarchy, *cities→functional zones→regions*. The hierarchical recurrent graph neural network can effectively capture dynamic long-range spatiotemporal dependency by propagating the information from top-level city to zone nodes and from zone nodes to bottom-level region nodes. In the third task, we introduce two tailor-designed self-supervised tasks, neighbor prediction and contextual inference, to exploit transferable knowledge in the spatial topology and the regions' contextual factors to regularize the model parameters for semi-supervised air quality forecasting.



(a) Distance v.s. air quality correlation.

(b) Temporal autocorrelation.

(c) Contextual similarity v.s. air quality correlation.

Fig. 4. Air quality correlations from spatial, temporal, and contextual view.

4 Methodology

In this section, we elaborate on the details of SSH-GNN.

4.1 Multi-View Air Quality Distribution Approximation

The real-time air quality condition is a strong signal for future prediction. However, the limited monitoring stations prevent us directly apply real-time air quality as a part of input feature. To this end, we propose a multi-view air quality distribution approximation module to estimate real-time air quality conditions for regions $\mathcal{R}_u$ from the *spatial view*, *temporal view*, and *contextual view*, simultaneously. The intuitions are three-fold. First, Fig. 4a depicts the air quality correlation (i.e. the Pearson coefficient) versus the geographical distance. As can be seen, the air quality of spatially adjacent regions are more related than distant ones. Second, Fig. 4b plots the temporal autocorrelation of air quality. We observe the air quality exhibit strong temporal dependency, such that the current air quality conditions also depend on previous air quality conditions. Third, Given a representative region $r_i \in \mathcal{R}$, Fig. 4c depicts the pair-wise relation between the regional contextual similarity and the air quality correlation. As expected, the regions with higher contextual similarities tend to have higher air quality correlations.

4.1.1 *Spatial View*

Consider the region air quality x_i^a of region $r_i \in \mathcal{R}_l$ at a particular time. We first estimate the current air quality from the spatial view. Specifically, we aim to learn a representation based on nearby regions with observed air quality,

$$\mathbf{x}_i^d = \sum_{j \in \mathcal{N}_s} s_{ij} \mathbf{W}_s \mathbf{x}_j^a, \tag{2}$$

where $\mathbf{x}_i^d$ is the estimated air quality representation from spatial domain, s_{ij} is proximity score between r_i and r_j, $\mathbf{W}_s$ is a weighted matrix we aim to learn, and $\mathcal{N}_s$ is the set of K nearest regions with observed air quality. We compute the proximity score s_{ij} based on a Gaussian kernel [15],

$$s_{ij} = exp\left(-\frac{dist(r_i, r_j)^2}{\delta^2}\right), \tag{3}$$

where $dist(r_i, r_j)$ is the geographical distance between region r_i and region r_j, and δ denotes the standard deviation of geographical distances.

4.1.2 *Temporal View*

We further estimate the air quality from temporal view by incorporating the local temporal information via a Gated Recurrent Unit (GRU) [16]. The implementation of GRU is as follows

$$\begin{cases} \mathbf{r}_i^t = \sigma(\mathbf{W}_r[\mathbf{h}_i^{t-1} \oplus \mathbf{x}_i^t] + \mathbf{b}_r) \\ \mathbf{z}_i^t = \sigma(\mathbf{W}_z[\mathbf{h}_i^{t-1} \oplus \mathbf{x}_i^t] + \mathbf{b}_z) \\ \widetilde{\mathbf{h}}_i^t = \tanh(\mathbf{W}_{\widetilde{h}}[\mathbf{r}_i^t \odot \mathbf{h}_i^{t-1} \oplus \mathbf{x}_i^t] + \mathbf{b}_{\widetilde{h}}) \end{cases}, \tag{4}$$

$$\mathbf{h}_i^t = \mathrm{GRU}(\mathbf{h}_i^{t-1}, \mathbf{x}_i^t) = (1 - \mathbf{z}_i^t) \odot \mathbf{h}_i^{t-1} + \mathbf{z}_i^t \odot \widetilde{\mathbf{h}}_i^t, \tag{5}$$

where W_r, W_z, $W_{\widetilde{h}}$, b_r, b_z, $b_{\widetilde{h}}$ are learnable parameters, $\oplus$ and $\odot$ denote the concatenation operation and the Hadamard product operation, respectively.

$$\mathbf{h}_i^t = \mathrm{GRU}(\mathbf{h}_i^{t-1}, x_i^d \oplus x_i^{m,t} \oplus x_i^{c,t}) \tag{6}$$

where $\mathbf{h}_i^{t-1}$ denotes the hidden state at time step $t-1$ from GRU, $\oplus$ is concatenate operation, $\mathbf{x}_i^m$ and $\mathbf{x}_i^c$ are weather and contextual features of region r_i, respectively. The temporal estimation is performed as follows

$$\mathbf{x}_i^e = \mathbf{W}_t \mathbf{h}_i^{t-1}, \tag{7}$$

where $\mathbf{x}_i^e$ is the estimated air quality representation from temporal domain, $\mathbf{W}_t$ is a learnable weighted matrix shared by all regions in the dataset. Note that we do not have real-time air quality information of the unmonitored regions. Therefore, the GRU model is learnt from data of the observed regions during the model training phase.

4.1.3 *Contextual View*

Additionally, we estimate current region air quality based on the semantic similarity of environmental contextual (e.g., green lands or industrial areas). Intuitively, urban contextual factors such as POI and road network density can be used to reflect the environmental context. Consider contextual features $\mathbf{x}_i^c$ and $\mathbf{x}_j^c$ of region r_i and r_j. We measure the semantic similarity between $\mathbf{x}_i^c$ and $\mathbf{x}_j^c$ by

$$c_{ij} = e^{-sim(\mathbf{x}_i^c, \mathbf{x}_j^c)}, \tag{8}$$

where $sim(\cdot,\cdot)$ is an euclidean distance based similarity function. The air quality can be estimated from regions with real-time air quality observations by the following contextual aggregator

$$\mathbf{x}_i^s = \sum_{j \in \mathcal{N}_c} c_{ij} \mathbf{W}_c \mathbf{x}_j^a, \tag{9}$$

where $\mathbf{x}_i^s$ is the estimated air quality representation from contextual domain, $\mathbf{W}_c$ is parameters aim to learn, and $\mathcal{N}_c$ is the set of K similar regions $r_j \in \mathcal{R}_l$.

4.1.4 Multi-View Estimation Fusion

We derive the unified embedding $\mathbf{x}_i^u$ and make final estimation by fusing all the above representations as follow

$$\mathbf{x}_i^u = \sigma(\text{AGGREGATE}(\{\mathbf{x}_i^d, \mathbf{x}_i^e, \mathbf{x}_i^s\})), \tag{10}$$

where σ is activation function (e.g., LeaklyReLU), $\text{AGGREGATE}(\cdot)$ denotes an aggregator function. In the experiment, we find that mean operator is more effective and efficient than other candidate aggregator functions, such as attention-based aggregator and sum operator. Thus, we choose mean operator function for the forecasting task. With the unified embedding $\mathbf{x}_i^u$, we estimate the current region air quality as a regression task

$$\hat{y}_i^r = \mathbf{w}_r \mathbf{x}_i^u, \tag{11}$$

where $\hat{y}_j^r$ is the estimated air quality values, $\mathbf{w}_r$ denotes learnable parameters. Additionally, to stabilize the numerical value forecasting [17], we further introduce a multi-class classification task to infer the current air quality index level, defined as

$$\hat{\mathbf{y}}_i^c = \text{Softmax}(\mathbf{w}_c \mathbf{x}_i^u), \tag{12}$$

where $\hat{\mathbf{y}}_i^c$ denotes the inferred air quality distribution. Note that to fully utilize the learned information in multi-view based approximation, we leverage $\mathbf{x}_i^u$ for subsequent forecasting task rather than the predicted scalar value.

4.2 Hierarchical Recurrent Graph Neural Network

Then we present the hierarchical recurrent graph neural network for dynamic long-range spatiotemporal dependency modeling. Specifically, we first construct the region graph at the bottom of the hierarchy. Then we can gradually form high-level clusters (i.e. cities, functional zones) from bottom to top by aggregating low-level units. In our model, we follow *cities→functional zones→regions* to build the hierarchical graph. Both regions and cities are real-world administrative divisions, while the functional zones are virtual nodes learned via POI and road network features. Since regions and cities are partitioned by administrative division, the corresponding adjacency matrices $\mathbf{A}^R$ and $\mathbf{A}^C$ can be directly calculated via the Gaussian kernel as defined in Equation (3), and $\mathbf{A}^{ZC}$ is automatically estimated during model training. By leveraging such a hierarchical graph, we can aggregate information at a high level and propagate useful long-range information to low-level regions, which encodes long-range dependencies among regions. We introduce the detailed spatial dependency modeling method from bottom to top.

4.2.1 Modeling Region Dependency

We adopt Graph Convolution Network (GCN) [18], as the basic building block for capturing spatial dependencies in the hierarchical region graph $\mathcal{G}^h$. GCN is a lightweight and effective GNN-based model, which can help reduce the computational overhead in hierarchical spatial dependency modeling. The detailed definition of the basic GCN operator, $\text{GConv}(\cdot)$ is as follows. Consider the input features of a particular typed node $\mathbf{X}$, we first define the graph convolution operation (*GConv*) as

$$\mathbf{X}' = \text{GConv}(\mathbf{X}, \mathbf{A}) = \sigma(\mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \mathbf{X} \mathbf{W}), \tag{13}$$

where $\mathbf{X}'$ is the updated node representation, $\mathbf{A}$ denotes the corresponding adjacency matrix with self-connection, $\mathbf{D}$ is degree matrix defined as $\mathbf{D} = \sum_j \mathbf{A}_{i,j}$, σ is non-linear activation function, $\mathbf{W}$ is a learned weighted matrix.

Based on the output of the multi-view learning module X^u, we first employ graph convolution to capture short-range dependencies by aggregating information from nearby regions

$$\mathbf{X}^r = \text{GConv}(\mathbf{X}^u \oplus \mathbf{X}^m \oplus \mathbf{X}^c, \mathbf{A}^R). \tag{14}$$

4.2.2 Modeling Functional Zone Dependency

In the real-world, each region may serve for several functionalities simultaneously. For example, the business district usually has many recreation facilities. Rather than clustering each region into a specific functional zone, we allow each region has a chance to belong to multiple functional zones with soft assignment probabilities. As found in previous studies [19], [20], various urban factors such as Point-of-Interests (POIs) and human activities can reflect the urban dynamics and functional patterns of a region. Therefore, we learn the soft assignment matrix $\mathbf{S}^{RZ}$ based on various contextual features via graph convolution,

$$\mathbf{S}^{RZ} = \text{GConv}(\mathbf{X}^c, \mathbf{A}^R), \tag{15}$$

where $\mathbf{S}^{RZ} \in \mathbb{R}^{N_R \times N_Z}$, each row measures the likelihood that a specific region is associated to different functional zones. Consider different spatiotemporal and functional distribution, we restrict each city has its own functional zones, which is not shared across different cities. Since functional zones may vary across cities, we allocate n_z independent functional zones for each city, where $N_Z = |\mathcal{C}| n_z$. We define a indication matrix $\mathbf{M}^{RZ}$, where $\mathbf{M}^{RZ}[r, z] = 1$ if region r and zone z belong to the same city and $\mathbf{M}^{RZ}[r, z] = 0$ otherwise. The *region-to-zone* assignment matrix $\mathbf{A}^{RZ}$ can be derived by

$$\mathbf{A}^{RZ} = \text{Softmax}(\mathbf{M}^{RZ} \odot \mathbf{S}^{RZ}), \tag{16}$$

where $\odot$ represents the element-wise product, each entry $\mathbf{A}^{RZ}[r, z]$ can be viewed as the probability of region r maps to functional zone z. The representation of each functional zone $\mathbf{x}_i^z \in \mathbf{X}^z$ can be derived using a linear combination of region representations

$$\mathbf{X}^z = \mathbf{A}^{RZ^\top} \mathbf{X}^u, \tag{17}$$

where $\mathbf{A}^{RZ^\top}$ is the transpose operation of $\mathbf{A}^{RZ}$.

Likewise, we further obtain the adjacency matrix $\mathbf{A}^Z$ by

$$\mathbf{A}^Z = \mathbf{A}^{RZ^\top} \mathbf{A}^R \mathbf{A}^{RZ}. \tag{18}$$

Similar to region-level spatial dependency modeling, we reuse the graph convolution operation to capture dependencies among different functional zones,

$$\mathbf{X}^{z'} = \mathrm{GConv}(\mathbf{X}^z, \mathbf{A}^Z). \tag{19}$$

To further integrate the influence of the regional effect of the air pollutant transport and dispersion process [21], we design a gating message passing mechanism to control information passing from functional zones to bottom regions

$$\mathbf{X}^{rz} = \mathbf{G}_z \odot (\mathbf{A}^{RZ}\mathbf{X}^{z'}), \tag{20}$$

where $\mathbf{G}_z$ is the output of gating mechanism, which is defined as

$$\mathbf{G}_z = \mathrm{Sigmoid}((\mathbf{X}^{m,t} \oplus \mathbf{X}^c)\mathbf{W}_z), \tag{21}$$

where $\mathbf{W}_z$ are learnable parameters, $\mathbf{X}^{m,t}$ and $\mathbf{X}^c$ are weather and contextual features (e.g., weather conditions and POI density). By using the above gating mechanism, we can shield irrelevant information from dissimilar regions and adaptively capture spatial interactions among distant regions under different contexts.

4.2.3 Modeling City Dependency

The city-level spatial dependency is modeled in a similar way as the functional zone. First of all, we calculate the soft assignment matrix $\mathbf{S}^{ZC}$ by using the same graph convolution operation as defined in Equation (15). To prevent the interference between the functional zones across different cities, we introduce a zone-to-city mapping matrix $\mathbf{M}^{ZC}$, where $\mathbf{M}^{ZC}[z,c]=1$ if zone z belongs to city c and $\mathbf{M}^{ZC}[z,c]=0$ otherwise. The *zone-to-city* assignment matrix $\mathbf{A}^{ZC}$ can be derived by

$$\mathbf{A}^{ZC} = \mathrm{Softmax}(\mathbf{M}^{ZC} \odot \mathbf{S}^{ZC}), \tag{22}$$

where $\odot$ represents the element-wise product, each entry $\mathbf{A}^{ZC}[z,c]$ can be viewed as the probability of functional zone z maps to city c. We further obtain the city representation $\mathbf{X}^p$ and $\mathbf{X}^{p'}$ by following the similar process of Equations (17) to (19). Then, we extend Equation (20) to propagate long-range dependency information from high-level city nodes to low-level region nodes

$$\mathbf{X}^{rc} = \mathbf{G}_c \odot (\mathbf{A}^{RZ}\mathbf{A}^{ZC}\mathbf{X}^{p'}), \tag{23}$$

$$\mathbf{G}_c = \mathrm{Sigmoid}((\mathbf{X}^{m,t} \oplus \mathbf{X}^c)\mathbf{W}_p). \tag{24}$$

Based on the region-level, zone-level and city-level representations of region r_i, $\mathbf{x}_i^{r,t} \in \mathbf{X}^r$, $\mathbf{x}_i^{rz,t} \in \mathbf{X}^{rz}$ and $\mathbf{x}_i^{rc,t} \in \mathbf{X}^{rc}$, we obtain the unified hierarchical representation as follows

$$\mathbf{x}_i^t = \mathbf{x}_i^{r,t} \oplus \mathbf{x}_i^{rz,t} \oplus \mathbf{x}_i^{rc,t}, \tag{25}$$

The final hierarchical representation $\mathbf{x}_i^t$ encodes long-range spatial dependency information in different spatial levels.

4.2.4 Modeling Temporal Dependency

The air quality of each region node are not only influenced by its neighbors in $\mathcal{G}^h$, but also depend on their previous status. Consider a region r_i and its past T step representations $\{\mathbf{x}_i^{t-T+1}, \ldots, \mathbf{x}_i^t\}$, where $\mathbf{x}_i^t$ is the output of the hierarchical graph neural network at time t. We use the GRU operation to capture temporal dependency among different time steps. We use $\mathbf{h}_i^{t-1}$ and $\mathbf{h}_i^t$ to denote the hidden states of r_i at time step $t-1$ and t, respectively.

The hidden state $\mathbf{h}_i^t$ encodes both the spatial and temporal dependencies of region r_i and can be utilized for region air quality forecasting. We employ a feed forward neural network $f(\cdot)$ to produce future air quality predictions

$$(\hat{y}_i^{t+1}, \hat{y}_i^{t+2}, \ldots, \hat{y}_i^{t+\tau}) = f(\mathbf{h}_i^t, \mathbf{x}_i^w), \tag{26}$$

where $\mathbf{x}_i^w$ denotes weather forecasting features.

4.3 Spatiotemporal Self-Supervision

So far, the above described model can be trained end-to-end to make predictions by minimizing the supervised loss based on sparse historical observations of regions with monitoring stations $\mathcal{R}_l$. However, the model learned in this way is biased toward regions $\mathcal{R}_l$, which will induce unsatisfied forecasting results for those unmonitored regions $\mathcal{R}_u$. Inspired by the recent success of self-supervised learning for improving model generalization capability [22], we propose two tailor-designed self-supervised learning tasks, neighbor prediction and contextual inference, to alleviate the data distribution discrepancy between the regions $\mathcal{R}_l$ and $\mathcal{R}_u$. The key insight of the self-supervision is to extract transferable knowledge hidden in rich topological and contextual data to improve the generalization ability of the learned model.

4.3.1 Neighbor Prediction

In the neighbor prediction task, we aim to encode structural properties of unmonitored regions into our model. The major idea of neighbor prediction is to leverage the spatial topological information as the supervision signal to optimize the region embeddings such that adjacent regions are close to each other in the latent space. Specifically, we first apply the multi-view learning module introduced in Section 4.1 to derive the ground-truth region embedding $\mathbf{x}_i^u$. Then we train a dedicated 1-layer graph convolution network (GCN) based on Equation (13), which derive the neighbor embedding $\mathbf{x}_i^{cx}$ by aggregating and transforming neighboring representations $\mathbf{x}_j^u$ for $r_j \in \mathcal{N}_i$. The objective of the neighbor prediction task is to optimize the cosine similarity between the current region embedding and its neighbor embeddings,

$$\mathcal{L}_p = -\log\left(\sigma(\mathbf{x}_i^{u\top}\mathbf{x}_i^{cx})\right) - \mathbb{E}_{j\sim P_n(i)}[\log\left(\sigma(-\mathbf{x}_i^{u\top}\mathbf{x}_j^{cx})\right)], \tag{27}$$

where $\sigma(\cdot)$ denotes Sigmoid function, $P_n(i)$ is a negative sampling distribution for region r_i, enabling us to randomly choose other regions as negative samples.

4.3.2 Contextual Inference

The contextual inference task aims to incorporate common knowledge of highly correlated predictive tasks by adopting various contextual features as supervision signals. In general, an arbitrary region attribute can be used as the self-supervision signal. However, in practice, the meteorological attributes are in a coarsen granularity (i.e. some adjacent nodes may share the same weather condition), and the POI and road network features are time-invariant. Predicting such less informative attributes may introduce additional noises and lead to negative transfer. To this end, we employ the historical air quality as self-supervision signals. Moreover, we adopt the traffic flow, which has been proven strongly correlated with air quality variation [13], as additional signals for contextual self-supervision. In particular, we randomly mask 5% of air quality and traffic flow features and then we infer the masked features based on Equation (11). After that, we obtain the loss of contextual inference $\mathcal{L}_q$ by summing *Mean Square Error* (MSE) $\mathcal{L}_{mse} = (\hat{x}_i^{mask} - x_i^{mask})^2$ of each masked feature, where $\hat{x}_i^{mask}$ and x_i^{mask} denote inferred values and ground truth of masked feature, respectively. In our implementation, the contextual inference component follows the multi-task learning paradigm with hard parameter sharing [17], such that different supervision signals back-propagate to the shared GCN network through task-specific output layers.

Finally, we combine the losses of neighbor prediction and contextual inference to obtain the self-supervised loss

$$\mathcal{L}_s = \mathcal{L}_p + \mathcal{L}_q. \tag{28}$$

4.4 Model Optimization

Following existing works [23], [24], we aim to minimize the MSE loss between the ground-truth observations and predictions,

$$\mathcal{L}_m = \frac{1}{\tau|\mathcal{R}_l|}\sum_{i=1}^{|\mathcal{R}_l|}\sum_{j=1}^{\tau}(\hat{y}_i^{t+j} - y_i^{t+j})^2. \tag{29}$$

Intuitively, we can achieve better future prediction if the model infer current air quality accurately for those unmonitored regions. To enhance the model prediction, we design an approximation module to infer real-time air quality values in Section 4.1. Therefore, we introduce two auxiliary losses to measure the estimation accuracy. For air quality regression task, we aim to minimize the error between the observed air quality and estimated air quality value in current time step t, the loss is defined as

$$\mathcal{L}_r = -\frac{1}{|\mathcal{R}_l|}\sum_{i=1}^{|\mathcal{R}_l|}(\hat{y}_i^{r,t} - y_i^{r,t})^2. \tag{30}$$

For the air quality classification task, we aim to minimize the error between the air quality level and estimated air quality distribution in current time step t, the *Cross-Entropy* (CE) loss is defined as

$$\mathcal{L}_c = -\frac{1}{|\mathcal{R}_l|}\sum_{i=1}^{|\mathcal{R}_l|}\mathbf{y}_i^{c,t}\log\hat{\mathbf{y}}_i^{c,t}. \tag{31}$$

In practice, the self-supervision tasks can be used to pretrain the network or jointly optimized with the main objective. In this work, we choose the second training strategy to reduce the model retraining time for handling the continuous air quality forecasting request. By considering the main task loss, auxiliary losses, and self-supervision losses, our model aims to jointly minimize the following objective

$$\mathcal{L} = \mathcal{L}_m + \lambda(\mathcal{L}_r + \mathcal{L}_c) + \beta\mathcal{L}_s, \tag{32}$$

where $\mathcal{L}_s$ is self-supervised loss, λ and β are the hyperparameters control the importance of auxiliary loss and self-supervision loss. Besides, we employ Adam optimizer [25] for training with exponential decay.

5 Experiments

We conduct extensive experiments on real-world data to evaluate: (1) the overall performance of SSH-GNN, (2) the ablation of each component in SSH-GNN, (3) the parameter sensitivity of SSH-GNN, (4) the robustness of our approach, and (5) the forecasting result visualization.

5.1 Datasets

We conduct experiments on two real-world datasets collected from Beijing-Tianjin-Hebei (BTH) and Pearl River Delta (PRD), two large urban agglomerations in China, for evaluation. Both datasets are ranged from January 1, 2018, to April 1, 2020. All the datasets include (1) air quality observations (i.e., AQI, PM2.5, PM10, O3, NO2, SO2, and CO), (2) weather observations and weather forecasting information (i.e., weather condition, temperature, humidity, pressure, wind speed and wind direction), and (3) urban contextual data (i.e., POI distribution, road network and urban traffic flows). All air quality observations are crawled from the China government website[1]. We associate POI, road network and real-time traffic flow to each region through open API provided by Baidu map [2]. Same as existing studies [4], our goal is to predict Air Quality Index (AQI) for each region, which is derived by the Chinese AQI standard. The statistics of two datasets are shown in Table 2.

- *Air quality data*. We collect six kinds of air pollutants reported by air quality monitoring stations, including PM2.5, PM10, O3, NO2, SO2, and CO. All air quality records are crawled from China government websites.
- *Meteorological data*. We collect meteorological data and weather forecast data from a public website, consisting of temperature, pressure, humidity, wind speed and wind direction every hour.
- *POI data*. Semantics in POIs indicate the function of regions and can directly affect the air quality. The POI data is collected from Baidu Map, one of the world's largest navigation applications. Each POI record has a POI name, category, and coordinates. We choose 20 informative POI categories for air quality prediction. Fig. 5 shows the POI distribution of each category.
- *Road network data*. The road network data is also derived from the Baidu Map. Road network data can capture the regional traffic capability. Each road segment contains a start location coordinates, an end location coordinates, the road length and the level of the road segment. There are eight levels of road segments.

[1] http://www.cnemc.cn/en/
[2] http://lbsyun.baidu.com/

TABLE 2
Statistics of Datasets

Data description	BTH	PRD
# of cities	11	9
# of regions	2405	686
# of air quality stations	479	57
# of air quality records	8,544,288	1,057,436
# of meteorological records	2,976,125	919,143
# of POIs	3,358,902	4,327,007
# of road segments	3,022,683	4,655,786
Average AQI	87.4	40.1

- *Traffic data*. We obtain country-wide traffic flows from our city profile platform every hour. Specifically, the real-time inflow and outflow of each region are calculated using massive trajectories and travel events collected from Baidu Map. Our traffic data contain the flows of five transport modes: taxi, drive, bus, walk, and cycle. We aggregate the first two transport modes as the car flow feature, the rest as the human flow feature.

5.2 Feature Construction

We construct various features from multi-source urban data for air quality prediction, including spatially-related and temporally-related features. For spatially-related features, we associate POI and road network to each region, and identify the following three features: (1) the POI count of each POI category; (2) the number of road segments and road intersections; (3) the total length of road segments. We use log transform to project the POI count features into the same order of magnitude, and further normalize the POI features of each region into a distribution. As lower POI density can promote the dispersion of air pollutants, we calculate the POI density, which is defined as the total number of POI divided by region area. For temporally-related features, we extract five types of features: (1) the air quality of the previous 12 hours at each observed region; (2) the weather conditions in the past 12 hours; (3) the weather forecast features of the time point we aim to predict; (4) the previous 12 hours' traffic flows; (5) hour, day of week, month, and holiday. The temporally-related features reflect the local meteorological and traffic patterns, which have great impact on future air quality. In addition, we transform the categorical features into one-hot vectors, and all the numerical features are normalized by Z-Score.

5.3 Experimental Settings

5.3.1 Hyperparameter Selection

Our model and all the deep learning baselines are implemented with PyTorch. All experiments are performed on a Linux server with 8 NVIDIA Tesla P40 GPUs. Each numerical feature is normalized by Z-score, and categorical features are embedded using a 4-dimensional vector. We set hierarchical graph convolutional layers $l = 1$. The dimension of hidden units and the cell size of GRU is set to 32. The functional zones of each city is fixed to $n_z = 12$. K, λ and β are set to 3, 0.5 and 0.1, respectively. We initialize model parameters with uniform distribution. The model is trained by Adam optimizer with a learning rate $lr = 0.0001$. We employ LeakyReLU (α=0.2) as activation function used in the hidden layers. We set $T = 12$ and the future time step $\tau = 24$.

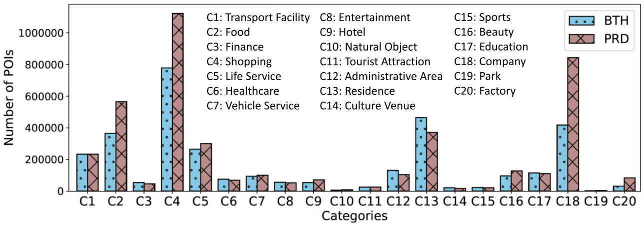


Fig. 5. POI distribution.

5.3.2 Data Split and Evaluation Metrics

Due to limited amount of regions with monitoring station, we follow the widely used setting [8], [9] for evaluation. We first randomly divide ground truth regions into three equal-sized parts. Then we run the model three times, and average the results as the final performance. Each time we select two parts as the training data, the rest part as unmonitored regions for evaluation. We use Mean Absolute Error (MAE) and Rooted Mean Square Error (RMSE) as evaluation metrics.

5.3.3 Baseline Methods

We compare the performance of our model with the following seven baselines. For a fair comparison, we carefully fine-tuned the hyper-parameters of each baseline on our datasets via grid search. Also note that all the baselines use the same input features as ours. As unmonitored regions do not have air quality observations, we use spatial interpolation [26] to estimate the real-time air quality, then we feed the estimated values together with other features as model inputs.

- *LR* uses linear regression for air quality prediction. To alleviate the data sparsity problem, we concatenate the observations 3 nearest air quality stations together with other features as model input.
- *GBRT* is a popular tree-based ensemble method. We use an effective version XGBoost [27], the input features is same as LR. Besides, we fix the minimal child weight to 3, and the learning rate and maximal tree depth are set to 0.1 and 5, respectively.
- *Seq2seq* leverages encoder-decoder architecture to predict air quality. We construct the seq2seq model by stacking two layers of GRU cells, and the cell size is set to 32.
- *DeepAir* [4] first employs a spatial transformation component to aggregate sparse air quality data, and then a distributed fusion network is adopted to capture the interactions among heterogeneous urban data. we set the number of hidden layers, embedding size and the dimension of hidden units to 9, 6, 64, respectively. The learning rate is fixed to 0.0001.
- *GeoMAN* [23] uses multi-level attention mechanism to capture spatiotemporal dependencies. External factors are fed in the decoder to enhance the prediction performance. The embedding size, the trade-off

TABLE 3
Semi-Supervised air Quality Forecasting Error Given by *MAE* and *RMSE* on BTH and PRD Datasets

Model	BTH (1-6/ 7-12/ 13-24 h)		PRD (1-6/ 7-12/ 13-24 h)	
	MAE	RMSE	MAE	RMSE
LR	36.53 / 40.28 / 50.16	49.23 / 60.81 / 66.44	26.49 / 30.57 / 33.86	35.41 / 40.65 / 45.37
GBRT	30.86 / 35.37 / 41.34	42.66 / 50.97 / 57.05	22.84 / 27.31 / 31.72	32.12 / 36.97 / 42.18
Seq2seq	28.21 / 32.47 / 36.28	39.84 / 48.19 / 53.58	18.13 / 24.85 / 28.65	30.83 / 34.79 / 41.07
GeoMAN	23.45 / 28.24 / 31.41	36.14 / 45.73 / 48.09	14.96 / 21.17 / 24.58	25.74 / 30.93 / 34.54
DeepAir	22.86 / 28.75 / 32.81	34.94 / 46.13 / 49.71	14.05 / 20.12 / 23.61	24.16 / 29.39 / 32.74
GC-DCRNN	22.42 / 27.31 / 30.64	34.25 / 43.58 / 47.01	13.46 / 18.23 / 22.18	22.72 / 28.64 / 31.58
MGED-Net	22.61 / 27.46 / 30.71	34.69 / 43.91 / 47.53	13.89 / 18.52 / 22.45	23.47 / 29.22 / 31.94
MasterGNN	21.59 / 26.86 / 30.27	33.82 / 42.97 / 46.44	13.01 / 17.95 / 21.89	21.92 / 27.87 / 30.76
SHARE	19.05 / 25.89 / 29.69	32.75 / 40.49 / 45.45	11.06 / 15.83 / 19.14	19.41 / 24.53 / 26.67
SSH-GNN (ours)	**16.43 / 22.59 / 26.37**	**28.51 / 36.64 / 41.26**	**9.72 / 13.64 / 16.17**	**17.08 / 20.95 / 23.41**

parameter, the cell size and the learning rate are respectively set to 8, 0.2, 64, and 0.0001.

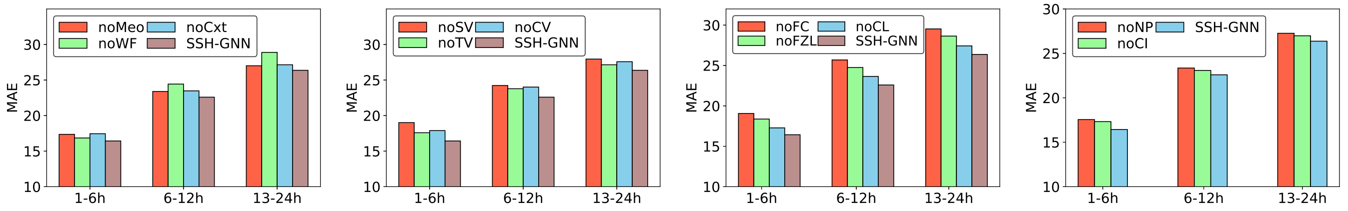
- *GC-DCRNN* [28] predicts air quality prediction by using a diffusion convolution recurrent neural network based on geographic features. We follow the graph construction method described in the original paper. The diffusion step is set to 2, the scheduled sampling probability is set to 0.5, and the learning rate is fixed to 0.0001.
- *MGED-Net* [24] first group features into multiple groups based on feature correlations, then a multi-group encoder-decoder network is used to fuse different feature groups for air quality prediction.
- *MasterGNN* [29] forecasts air quality by leveraging graph neural network and multi-adversarial learning. The interactions between air quality and weather conditions can be explicitly modeled via this framework.
- *SHARE* [30] is an advanced semi-supervised deep learning framework for parking availability prediction. We implement the model following the original paper. The dimension of GRU is set to 32, the number of latent nodes is set to 200. Besides, ϵ, k, and λ are fixed to 20, 3, 0.5, respectively, with 0.0001 learning rate.

5.4 Baseline Parameter Settings

This section details the implementation and hyper-parameter setting of each baseline. For baselines, we carefully finetune hyper-parameters via grid search. To utilize the sparse air quality, we adopt inverse distance weighting to infer the real-time air quality for each unmonitored region. After that, we concatenate the inferred values air quality with other features (e.g., POI, road network, and traffic flow features) as model input. For LR, we first feed all the features into the model, then train the model parameters together with L1 regularization. For GBRT, we use an effective version XGBoost. Besides, we fix the minimal child weight to 3, and the learning rate and maximal tree depth are set to 0.1 and 5, respectively. For Seq2seq, we construct the seq2seq model by stacking 2 layers of GRU cells, and the cell size is set to 32. For DeepAir, we set the number of hidden layers, embedding size and the dimension of hidden units to 9, 6, 64, respectively. The learning rate is fixed to 0.0001. For GeoMAN, we train models for each city to reduce the computation complexity of attention mechanism. The embedding size, the trade-off parameter, the cell size and the learning rate are respectively set to 8, 0.2, 64, and 0.0001. For GC-DCRNN, we follow the graph construction method described in the original paper. The diffusion step is set to 2, the scheduled sampling probability is set to 0.5, and the learning rate is fixed to 0.0001. For MGED-Net, we group different feature following the objective function in the original paper. The dropout rate and the cell size of the encoder and decoder are fixed to 0.3 and 16. We also set the learning rate to 0.0001. Besides, the optimal number of groups is 4. For MasterGNN, we set graph attention layers $l = 2$. The cell size of the GRU is set to 32. We set distance threshold to 15, and the hidden size of the multi-layer perceptron of each discriminator is fixed to 32. Finally, for SHARE, we implement the model following the original paper. The dimension of GRU is set to 32, the number of latent nodes is set to 200. Besides, ϵ, k, and λ are fixed to 20, 3, 0.5, respectively, with 0.0001 learning rate.

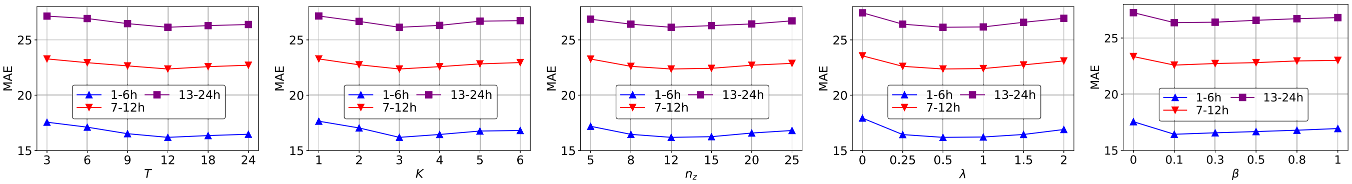
5.5 Overall Performance

Table 2 presents the overall results of SSH-GNN and compared baselines on two datasets with respect to MAE and RMSE. As we can see, our approach consistently outperforms all the baselines using both metrics, which demonstrates the superiority of SSH-GNN. Specifically, our model achieves (15.9%, 14.6%, 12.6%) and (14.9%, 10.5%, 10.2%) improvements beyond the best baseline (SHARE) on MAE and RMSE on BTH dataset for (1-6h, 7-12h, 12-24h) prediction, respectively. Similarly, the improvement of MAE and RMSE on PRD dataset are (13.8%, 16.1%, 18.4%) and (13.6%, 17.1%, 13.9%), respectively. Going deep into these approaches, we can observe LR and GBRT have the worst performance, since they cannot handle complex spatiotemporal dependencies. All GNN based approaches (GC-DCRNN, MasterGNN, SHARE, SSH-GNN) outperform other baseline methods, which indicates the effectiveness of modeling non-euclidean spatial structure for air quality forecasting. Moreover, MasterGNN perform better than GC-DCRNN by utilizing the advantage of adversarial learning. Among these baselines, SHARE achieves the best performance, demonstrating the advantage of semi-supervised learning paradigm. Besides, we notice that both metrics on PRD dataset are relatively smaller than on BTH dataset. The possible reason is that BTH suffers from more serious air pollution than PRD, and have more complicated environmental



(a) Effect of different feature sets. (b) Effect of multi-view air quality distribution approximation. (c) Effect of hierarchical recurrent graph neural network. (d) Effect of spatiotemporal self-supervision.

Fig. 6. Ablation study of SSH-GNN on the BTH dataset.



(a) Effect of T. (b) Effect of K. (c) Effect of n_z. (d) Effect of λ. (e) Effect of β.

Fig. 7. Parameter sensitivities on the BTH dataset.

contextual factors (e.g., more heavy industry factories) influencing the air quality.

5.6 Ablation Study

Then we verify the effectiveness of each component in SSH-GNN. Due to the page limit, we only report the results on BTH dataset by using MAE. The results on BTH dataset using RMSE and on PRD dataset using both metrics are similar.

Effect of Different Feature Sets. We first verify the features used in the proposed model. Specifically, we evaluate three variants of SSH-GNN: (1) *noMeo* removes meteorological features, (2) *noWF* removes weather forecasting features, (3) *noCxt* removes contextual features. As shown in Fig. 6a, we observe a performance gain by adding different types of features. In particular, weather forecast features have a significant impact on long-term prediction. The above results demonstrate the effectiveness of alleviating data sparsity problem using multi-source urban data compared with simply using air quality data as the model input.

Effect of Multi-View air Quality Distribution Approximation. We compare three variants of SSH-GNN, (1) *noSV* excludes the spatial view estimation, (2) *noTV* excludes temporal view estimation, (3) *noCV* excludes contextual view estimation. As depicted in Fig. 6b, SSH-GNN consistently outperforms *noSV*, *noTV*, and *noCV* by aggregating information from both spatial, temporal and contextual domains, demonstrating the advantage of incorporating multiple views approximation for semi-supervised air quality prediction.

Effect of Hierarchical Recurrent Graph Neural Network. We compare the following variants: (1) *noZCL* without either functional zones and cities, (2) *noFZL* without functional zones, (3) *noCL* without cities. We remove all the component related to specified kind of nodes during training. For instance, the model without functions zones only construct the graph following the hierarchy *cities→regions*. As shown in Fig. 6c, we observe that *noFZL* perform better than *noCL*, which demonstrates the importance of modeling spatial dependencies among functional zones. SSH-GNN achieves a better performance than *noZCL*, *noFZL*, and *noCL*, demonstrating the advantage of SSH-GNN in capturing long-range spatial dependencies at different levels.

Effect of Spatiotemporal Self-Supervision. To validate the effect of spatiotemporal self-supervision, we exam variants including (1) *noNP* removes neighbor prediction, and (2) *noCI* without contextual inference. As shown in Fig. 6d, we observe performance degradation when only involve one self-supervised task, either neighbor prediction or contextual inference. Overall, appropriate self-supervised learning tasks can incorporate transferable knowledge to improve model generalizability and lead to a better performance.

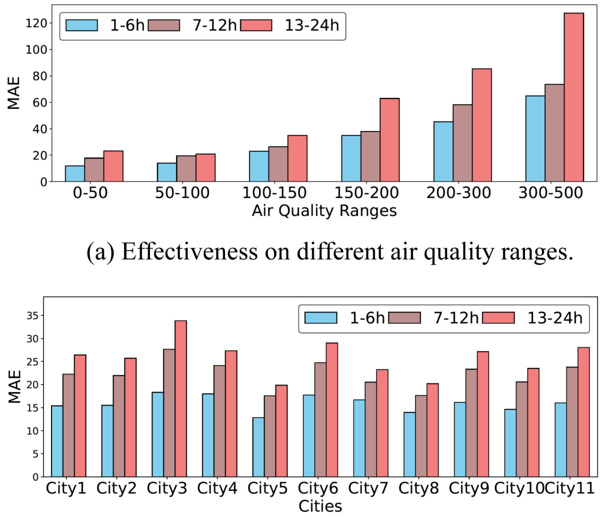
5.7 Parameter Sensitivity

We further study the parameter sensitivity of SSH-GNN. We report MAE on the BTH dataset. Each time we vary a parameter, we set others to their default values.

First, we vary the input length T from 3 to 24. The results are reported in Fig. 7a. We observe a performance gain when increasing T from 3 to 12, but a performance degradation by further increasing T from 12 to 24. The possible reason is that a large T incorporates too much redundant temporal information, which introduces extra noises for future prediction.

Then, we vary the number of neighbors K in multi-view multi-task estimator from 1 to 6, shown in Fig. 7b. As the K increases, the performance first increases and then gradually decreases. The reason perhaps is a small K can not provide enough neighborhood information, whereas too large K may introduce noises for learning air quality distribution and leading to performance degradation.

After that, to test the impact of the number of functional zones n_z in each city, we vary n_z from 5 to 25. The results are reported in Fig. 7c. Overall, our model achieves the best performance when $n_z = 12$. We observe consistent performance degradation when decrease or increase n_z.



(a) Effectiveness on different air quality ranges.

(b) Effectiveness on different cities.

Fig. 8. Effectiveness on Different Subgroups.

To test the impact of the auxiliary task weight λ, we vary λ from 0 to 2. The results are reported in Fig. 7d. We observe that the performance increases rapidly when varying λ from 0 to 0.5. But the performance degrades by further increasing λ. The main reason is too large λ makes the model focus on auxiliary tasks and weakens the importance of the main forecasting task.

Finally, we vary the self-supervised task weight β from 0 to 1. The results are reported in Fig. 7e. We can observe that the model achieve the best result when $\beta = 0.1$. Then the performance drop slightly by further increasing β from 0.1 to 1. Thus, we set β to 0.1 in our model to obtain the optimal performance.

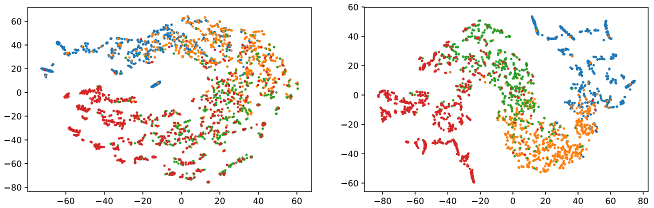
5.8 Effectiveness on Different Subgroups

In this section, we study the robustness of SSH-GNN on different subgroups, including: (1) group by different air quality ranges and (2) group by different cities. First, we group air quality by several consecutive ranges. i.e. less than 50, 50 to 100, 100 to 150, 150 to 200, 200 to 300, and more than 300. Fig. 8a shows the results of SSH-GNN on different air quality ranges on BTH dataset. We observe the performance degrades when the air quality goes large. Since we do not have sufficient training samples larger than 300, the performance drops rapidly in the range of 300-500. Furthermore, to test the effectiveness on different cities, we split testing instances by cities. Fig. 8b shows the results of SSH-GNN on BTH dataset. We observe city 3 (Shijiazhuang) is the most influential city to the overall performance. This is possibly because Shijiazhuang is the most polluted city in Beijing-Tian-Hebei. In future, optimize our model for these worse performed subgroups can help improve the overall performance.

5.9 Visualization

In this part, we qualitatively analyze why our approach can yield better performance for semi-supervised air quality forecasting.

We first qualitatively analyze why self-supervised learning can yield better performance on semi-supervised air quality forecasting task. Here we visualize the learned air quality representations with and without self-supervised learning. Specifically, we randomly select 4000 samples for visualization and leverage t-SNE [31] to project the learned representations into a 2-dimensional space. Each air quality embedding is colored based on their corresponding air quality ranges (1-50, 50-100, 100-200, 200-300), as depicted in Fig. 9. Without self-supervised learning, we observe the learned air quality embeddings in different ranges are mixed together without clear class boundaries. In contrast, we can see the learned embeddings by SSH-GNN in different air quality ranges are well classified with clear boundaries. The above observations further validate the benefit of our spatiotemporal self-supervised learning strategies on improving the discriminative capability of air quality embeddings.

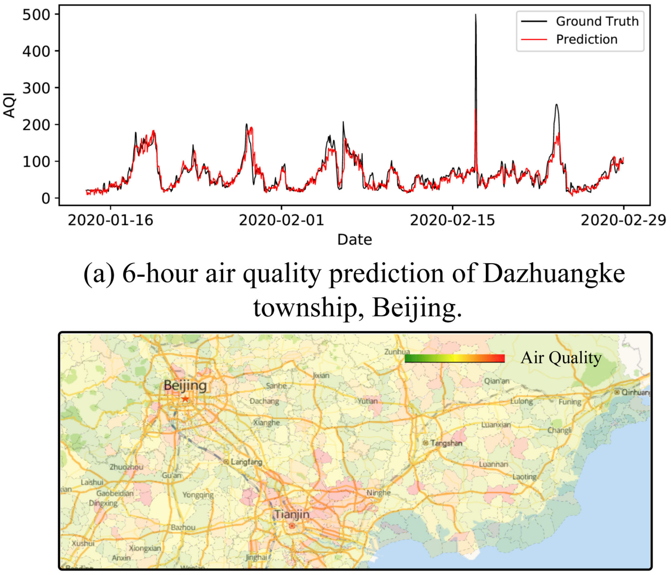


(a) Embeddings without self-supervised learning.

(b) Embeddings with self-supervised learning.

Fig. 9. Visualization embedding on BTH dataset.

Then we conduct visualizations to further demonstrate the effectiveness of our approach. Fig. 10a presents the prediction results of the Dazhuangke area in Beijing from Jan. 16, 2020 to Feb. 29, 2020. The black line denotes the ground truth air quality observed by monitoring stations, and the red line plots the predicted values obtained using SSH-GNN. As can be seen, two lines are very close and show similar trends, demonstrating our model accurately captured air quality variation patterns. Fig. 10b shows an example spatial air quality distribution predicted by our method at a specific time slot. We can observe the air quality of two



(a) 6-hour air quality prediction of Dazhuangke township, Beijing.

(b) Air quality heatmap predicted by SSH-GNN.

Fig. 10. Visualization.

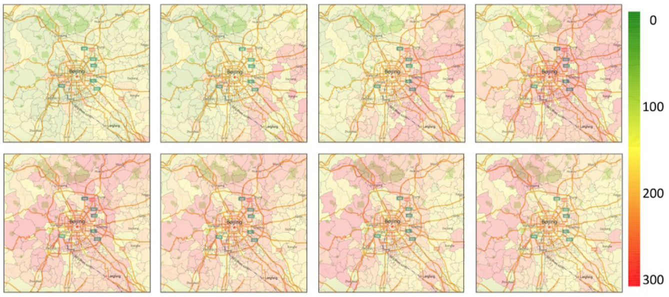


Fig. 11. Air quality heat maps in Beijing.

regions can be largely different though they are spatially close. Moreover, we observe more series pollution in downtown areas than in the suburbs.

Finally, Fig. 11 depicts the air quality changes of different time stamps in Beijing. The first four pictures are previous spatial air quality distribution estimated by multi-view learning module, the last four pictures are future heat maps predicted by our model. During this time, we find that the air pollution starts from the southeast of the city, then transports to the northwest and cover the whole city. The major reason is that there are many heavy industrial factories locate in the southeast, and a southeast wind carries the pollutants to the downtown area. Based on visualization analysis, we can help government administrators to identify the potential pollution regions in the future to help air pollution risk assessment. Such knowledge can impact authorities' decision making on pollution emission controls and early interventions. Since mid 2020, an interactive system has been deployed in the Ministry of Ecology and Environmental of China, serving for fine-grained air quality forecasting and environmental risk assessment for three urban agglomerations in China, i.e. Beijing-Tianjin-Hebei (BTH), Pearl River Delta (PRD), and Yangtze River Delta (YRD).

6 Related Work

We briefly review related literature, including air quality prediction and graph neural network.

6.1 Air Quality Forecasting

Existing studies on air quality forecasting mainly fall into two categories: numerical-based models and learning-based models. Numerical-based models make predictions by simulating the dispersion of air pollutants based on physical laws [32], [33]. However, insufficient pollutant sources information would induce strong biases in forecasting air quality. Besides, physical models are usually computationally expensive. With the availability of massive historical data (e.g., air quality observations and heterogeneous urban data), learning-based models have gained increasing attention. Statistical models such as ARIMA [34] and SVM [35] are widely used in the earlier study, but fail to capture complex spatiotemporal dependencies from the historical data. Zheng *et al.* [3] propose a hybrid predictive model to forecast air quality from both spatial and temporal perspectives. Recently, deep learning techniques have been applied to solve various spatiotemporal data mining tasks [36], [37], [38], [39], [40], they all achieve better performance beyond classical methods by a large margin. Many deep learning models [4], [5], [6], [23], [24], [28], [41], [42] have also been proposed to enhance the performance of air quality prediction. To name a few, DeepAir [4] fuses embeddings of various urban factors for air quality prediction. GC-DCRNN [28] employs a geo-context based diffusion convolutional recurrent neural network to model the dispersion of air pollutants. By leveraging the representation capacity of deep neural network, learning-based models usually achieve better prediction performance than numerical-based models. Different with previous works focus on station-level or city-level air quality prediction, in this paper we study the problem of semi-supervised fine-grained region-level air quality prediction, especially for these regions without monitoring stations deployed.

6.2 Graph Neural Network

Graph neural networks (GNNs) aims to extend deep neural network to deal with non-euclidean graph structure, which can be roughly categorized into two classes: spectral-based methods [18], [43], [44], [45] and spatial-based methods [46], [47], [48], [49]. On the one hand, spectral-based methods introduce the graph signal processing techniques to the deep learning domain. On the other hand, spatial-based methods directly define graph convolution operation based on spatial relations of different nodes. For example, Graph attention network (GAT) [47] leverages a self-attention mechanism to select important neighbors adaptively and then aggregate them with different weights. Due to its effectiveness, GNN has been widely applied in many urban computing tasks, such as traffic flow prediction [15] and parking availability prediction [30]. Recently self-supervised learning on graphs [22], [50], [51] has gained much attention. They show that the GNNs can achieve better generalization via various self-supervised tasks. For instance, Hu *et al.* [22] present several self-supervised learning strategies for pre-training GNNs at the level of the individual node and entire graphs. Qiu *et al.* [50] adopt contrastive learning to pre-train a graph neural network. In this work, we propose a hierarchical recurrent graph neural network structure with tailor-designed self-supervision tasks for semi-supervised air quality forecasting.

7 Conclusion

In this paper, we presented SSH-GNN, a semi-supervised region-level air quality forecasting framework based on sparse historical air quality observations and multi-sourced urban data. Specifically, we first proposed a multi-view learning method to estimate air quality distribution for regions without monitoring station deployed. Then, we developed a hierarchical recurrent graph neural network to model dynamic long-range dependencies between distant regions and utilize contextual features to incorporate the functional characteristics. Besides, two self-supervised learning auxiliary tasks are introduced to alleviate distribution discrepancy by integrating both structural and contextual information. Finally, extensive experimental results on two real-world datasets demonstrate that the performance of SSH-GNN consistently outperforms seven baselines.

References

[1] D. Campbell-Lendrum and A. Prüss-Ustün, "Climate change, air pollution and noncommunicable diseases," *Bull. World Health Org.*, vol. 97, no. 2, 2019, Art. no. 160.

[2] K. Matus, K.-M. Nam, N. E. Selin, L. N. Lamsal, J. M. Reilly, and S. Paltsev, "Health damages from air pollution in china," *Glob. Environ. Change*, vol. 22, no. 1, pp. 55–66, 2012.

[3] Y. Zheng *et al.*, "Forecasting fine-grained air quality based on big data," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 2267–2276.

[4] X. Yi, J. Zhang, Z. Wang, T. Li, and Y. Zheng, "Deep distributed fusion network for air quality prediction," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 965–973.

[5] Z. Luo, J. Huang, K. Hu, X. Li, and P. Zhang, "AccuAir: Winning solution to air quality prediction for KDD cup 2018, " in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1842–1850.

[6] X. Yi, Z. Duan, R. Li, J. Zhang, T. Li, and Y. Zheng, "Predicting fine-grained air quality based on deep neural networks," *IEEE Trans. Big Data*, early access, Dec. 24, 2020, doi: 10.1109/TBDATA.2020.3047078.

[7] S. Wang, Y. Li, J. Zhang, Q. Meng, L. Meng, and F. Gao, "PM2. 5-GNN: A domain knowledge enhanced graph neural network for PM2. 5 forecasting," 2020, *arXiv:2002.12898*.

[8] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-Air: When urban air quality inference meets big data," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 1436–1444.

[9] W. Cheng, Y. Shen, Y. Zhu, and L. Huang, "A neural attention model for urban air quality inference: Learning the weights of monitoring stations." in *Proc. 32nt AAAI Conf. Artif. Intell.*, 2018, pp. 2151–2158.

[10] W. R. Tobler, "A computer movie simulating urban growth in the detroit region," *Econ. Geogr.*, vol. 46, no. sup1, pp. 234–240, 1970.

[11] R. Ignaccolo, S. Ghigo, and S. Bande, "Functional zoning for air quality," *Environ. Ecological Statist.*, vol. 20, no. 1, pp. 109–127, 2013.

[12] Y. Wang, Y. Li, Z. Qiao, and Y. Lu, "Inter-city air pollutant transport in the beijing-tianjin-hebei urban agglomeration: Comparison between the winters of 2012 and 2016, " *J. Environ. Manage.*, vol. 250, 2019, Art. no. 109520.

[13] C. Zhang, Y. Zheng, X. Ma, and J. Han, "Assembler: Efficient discovery of spatial co-evolving patterns in massive geo-sensory data," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 1415–1424.

[14] H. Chen and R. Copes, *Review of Air Quality Index and Air Quality Health Index: Environmental and Occupation Health*. Toronto, Ontario: Public Health Ontario, 2013.

[15] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.

[16] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[17] Y. Zhang and Q. Yang, "A survey on multi-task learning," 2017, *arXiv:1707.08114*.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[19] J. Yuan, Y. Zheng, and X. Xie, "Discovering regions of different functions in a city using human mobility and pois," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2012, pp. 186–194.

[20] N. Wu, X. W. Zhao, J. Wang, and D. Pan, "Learning effective road network representation with hierarchical graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 6–14.

[21] A. Ding, X. Huang, and C. Fu, "Air pollution and weather interaction in east asia," in *Oxford Research Encyclopedia of Environmental Science*, London, U.K.: Oxford Univ. Press, 2017.

[22] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," in *Proc. Int. Conf. Learn. Representations*, 2020.

[23] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "GeoMAN: Multi-level attention networks for Geo-sensory time series prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3428–3434.

[24] Y. Zhang *et al.*, "Multi-group encoder-decoder networks to fuse heterogeneous data for next-day air quality prediction," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4341–4347.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[26] D. W. Wong, L. Yuan, and S. A. Perlin, "Comparison of spatial interpolation methods for the estimation of air quality data," *J. Exposure Sci. Environ. Epidemiol.*, vol. 14, no. 5, pp. 404–415, 2004.

[27] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. knowl. Discov. Data Mining*, 2016, pp. 785–794.

[28] Y. Lin *et al.*, "Exploiting spatiotemporal patterns for accurate air quality forecasting using deep learning," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geogr. Informat. Syst.*, 2018, pp. 359–368.

[29] J. Han, H. Liu, H. Zhu, H. Xiong, and D. Dou, "Joint air quality and weather prediction based on multi-adversarial spatiotemporal networks," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 4081–4089.

[30] W. Zhang, H. Liu, Y. Liu, J. Zhou, and H. Xiong, "Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 1186–1193.

[31] L. V. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[32] S. Vardoulakis, B. E. Fisher, K. Pericleous, and N. Gonzalez-Flesca, "Modelling air quality in street canyons: A review," *Atmospheric Environ.*, vol. 37, no. 2, pp. 155–182, 2003.

[33] N. K. Arystanbekova, "Application of gaussian plume models for air pollution simulation at instantaneous emissions," *Math. Comput. Simul.*, vol. 67, no. 4/5, pp. 451–458, 2004.

[34] L. A. Díaz-Robles *et al.*, "A hybrid ARIMA and artificial neural networks model to forecast particulate matter in urban areas: The case of temuco, chile," *Atmospheric Environ.*, vol. 42, no. 35, pp. 8331–8340, 2008.

[35] L. Wang and Y. P. Bai, "Research on prediction of air quality index based on NARX and SVM," *Appl. Mechanics Mater.*, vol. 602, pp. 3580–3584, 2014.

[36] J. Hu, B. Yang, C. Guo, C. S. Jensen, and H. Xiong, "Stochastic origin-destination matrix forecasting using dual-stage graph convolutional, recurrent neural networks," in *Proc. IEEE 36th Int. Conf. Data Eng.*, 2020, pp. 1417–1428.

[37] J. Hu, C. Guo, B. Yang, and C. S. Jensen, "Stochastic weight completion for road networks using graph convolutional networks," in *Proc. IEEE 35th Int. Conf. Data Eng.*, 2019, pp. 1274–1285.

[38] B. Zheng *et al.*, "SOUP: Spatial-temporal demand forecasting and competitive supply," *IEEE Trans. Knowl. Data Eng.*, early access, Sep. 14, 2021, doi: 10.1109/TKDE.2021.3110778.

[39] W. Zhang *et al.*, "Intelligent electric vehicle charging recommendation based on multi-agent reinforcement learning," in *Proc. Web Conf.*, 2021, pp. 1856–1867.

[40] J. Han, Y. He, J. Liu, Q. Zhang, and X. Jing, "GraphConvLSTM: Spatiotemporal learning for activity recognition with wearable sensors," in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–6.

[41] S. Du, T. Li, Y. Yang, and S.-J. Horng, "Deep air quality forecasting using hybrid deep learning framework," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2412–2424, Jun. 2021.

[42] Z. Chen, H. Yu, Y.-A. Geng, Q. Li, and Y. Zhang, "EvaNet: An extreme value attention network for long-term air quality prediction," in *Proc. IEEE Int. Conf. Big Data*, 2020, pp. 4545–4552.

[43] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Informat. Process. Syst.*, 2016, pp. 3844–3852.

[44] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3546–3553.

[45] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "CayleyNets: Graph convolutional neural networks with complex rational spectral filters," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 97–109, Jan. 2019.

[46] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Informat. Process. Syst.*, 2017, pp. 1024–1034.

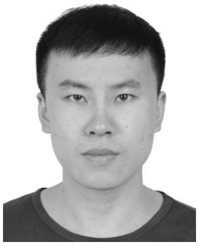
[47] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[48] H. Gao, Z. Wang, and S. Ji, "Large-scale learnable graph convolutional networks," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1416–1424.

[49] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proc. Adv. Neural Informat. Process. Syst.*, 2018, pp. 4800–4810.

[50] J. Qiu *et al.*, "GCC: Graph contrastive coding for graph neural network pre-training," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 1150–1160.

[51] Y. You, T. Chen, Z. Wang, and Y. Shen, "When does self-supervision help graph convolutional networks?," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10871–10880.



Jindong Han received the master's degree in information and communication engineering from the Beijing University of Posts and Telecommunications in 2020. He is currently working toward the PhD degree with the Hong Kong University of Science and Technology. His research interests mainly include data mining and machine learning.

Hao Liu received the BE degree from the South China University of Technology (SCUT) in 2012 and the PhD degree from the Hong Kong University of Science and Technology (HKUST), in 2017. He is currently an assistant professor with Artificial Intelligence Thrust, HKUST. Prior to that, he was a senior research scientist with Baidu Research and a postdoctoral fellow with HKUST. He has authored or coauthored prolifically in refereed journals and conference proceedings, such as TKDE, KDD, SIGIR, WWW, AAAI, and IJCAI. His general research interests include data mining, machine learning, and big data management, with a special focus on mobile analytics and urban computing.

Haoyi Xiong (Senior Member, IEEE) received the PhD degree in computer science from Telecom SudParis, Université Pierre et Marie Curie, Paris, France, in 2015. From 2016 to 2018, he was a tenure-track assistant professor with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA (formerly known as University of Missouri at Rolla). From 2015 to 2016, he was a research associate with the Department of Systems and Information Engineering, University of Virginia, Charlottesville, VA, USA. In 2018, he joined Big Data Laboratory, Baidu Research, Beijing, China, as a staff R&D engineer and research scientist, where he is currently a principal R&D architect and research scientist. He also holds an honorary appointment as a graduate faculty scholar affiliated to the ECE PhD Program with the University of Central Florida, Orlando, FL, USA. He has authored or coauthored more than 70 papers in top computer science conferences and journals, such as ICML, KDD, UbiComp, ICLR, RTSS, AAAI, IJCAI, ICDM, PerCom, *IEEE Internet of Things Journal*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Computers*, *ACM Transactions on Intelligent Systems and Technology*, and *ACM Transactions on Knowledge Discovery from Databases*. His current research interests include automated deep learning (AutoDL), ubiquitous computing, artificial intelligence, and cloud computing. He gave keynote speak in a series of academic and industrial activities, such as the industrial session of the 19th IEEE International Conference on Data Mining (ICDM'19), and was the poster co-chair of the 2019 IEEE International Conference on Big Data (IEEE Big Data'19). Dr. Xiong was the recipient of the Best Paper Award from IEEE UIC 2012, Outstanding Ph.D. Thesis Runner Up Award from CNRS SAMOVAR 2015, IEEE TCSC Award for Research Excellence (Early Career Researchers) 2020, and the co-recipient of the Science & Technology Advancement Award from the Chinese Institute of Electronics 2019. Many of his research outcomes in Baidu have been contributed to the open-source deep learning framework.

Jing Yang received the BS degree in environmental science from Nanjing University, Nanjing, China, in 2012, the MS degree in environmental science and engineering from Guangxi University, Nanning, China, in 2015, and the PhD degree in environmental science and engineering from Nanjing University, in 2019. Since 2019, she has been an engineer with the Environmental Development Center of the Ministry of Ecology and Environment. Her research interests include Big data technology and application, environmental economics, and environmental management.