GY485 Dissertation

Predicting Roadside Emissions Using Spatiotemporal Graph Neural Networks: A Case Study in London

Candidate Number:

Department of Geography and Environment

London School of Economics and Political Science

Word Count: 9784

CONTENT

Abstract

This study addresses the challenges of predicting road emissions, a major contributor to urban air pollution, by proposing a novel spatiotemporal graph neural network model. Traditional models, which often rely on physics-based deterministic approaches, struggle with computational intensity and limited generalization, while more recent data-driven deep learning models excel in capturing nonlinear processes. However, these models often overlook the critical spatial characteristics of the geographic environment. To overcome these limitations, our model effectively integrates Long Short-Term Memory (LSTM) for temporal dependency modeling with graph neural networks (GNN) to capture complex spatial relationships. By leveraging multi-source data, including meteorological conditions, land use, and traffic characteristics, the model dynamically adjusts spatial correlations between monitoring stations based on wind speed and direction. This approach enhances the model's ability to accurately predict road emissions across different time intervals and locations, providing valuable insights for urban planning and pollution control. The proposed model demonstrates superior performance in capturing the spatiotemporal dynamics of road emissions compared to existing methods.

1 Introduction

Road emissions are a major source of urban air pollution, particularly in areas with heavy traffic. The combustion of fossil fuels by vehicles releases harmful substances such as nitrogen oxides (NOx), particulate matter (PM), carbon monoxide (CO), and volatile organic compounds (VOCs), which directly impact air quality and pose significant health risks to the public. Prolonged exposure to high concentrations of these pollutants increases the risk of respiratory diseases, cardiovascular conditions, and even cancer (Xu et al., 2019). Additionally, road emissions exacerbate the greenhouse effect and contribute to climate change, affecting the balance of the ecosystem. Therefore, controlling and reducing road emissions has become an urgent task to protect the environment and safeguard public health.

As one of the largest cities in the world, London experiences intense traffic, with a dense network of roads, making road emissions a significant factor affecting air quality. These

pollutants primarily originate from vehicle exhaust, especially from diesel vehicles, heavy trucks, and buses. The congestion in central London leads to frequent vehicle starts, stops, and idling, resulting in the direct release of harmful substances into the streets and residential areas, severely impacting local air quality and public health. In recent years, the London government has implemented various measures to address this issue, such as introducing the Ultra Low Emission Zone (ULEZ) to charge high-emission vehicles entering the city center, promoting clean energy buses, and encouraging green travel options (Prieto-Rodriguez et al., 2022). However, with the continuous expansion of the city and the increasing number of vehicles, London's road emission problem remains severe, requiring ongoing monitoring and further action to improve air quality and residents' living conditions. The London government and relevant agencies have invested significant resources in monitoring and controlling roadside emissions. Numerous air quality monitoring stations are installed along major streets and intersections, recording real-time variations in roadside pollutant concentrations, which helps researchers and policymakers understand the spatial distribution and temporal dynamics of pollutants (Hood et al., 2018). The data not only reflects the direct impact of vehicle emissions but also reveals the significant influence of weather conditions (such as wind speed, temperature, and humidity) and traffic patterns (such as peak and off-peak hours) on roadside pollutant concentrations.

Spatiotemporal prediction modeling of road emissions holds substantial practical significance for a metropolis like London. First, spatiotemporal prediction modeling can help identify and analyze the spatiotemporal distribution characteristics of pollutants. By integrating multi-source data such as historical emission records, traffic volume, meteorological conditions, and land use, the model can predict emissions during different time periods (e.g., peak and off-peak hours) and at various locations (e.g., major thoroughfares and residential side streets). This allows researchers and policymakers to anticipate NO2 concentration peaks at specific times and locations, enabling more precise and effective interventions, such as preemptive traffic control or adjusting traffic signal timings to reduce pollutant accumulation and dispersion. At a broader level, in urban planning and road traffic design, planners can promote urban greening, use natural

barriers to absorb pollutants, or optimize road layouts to alleviate traffic congestion, thereby reducing the generation and accumulation of NO2.

Various predictive models have been developed to track pollutant concentrations generated by roadside vehicle emissions. Early models were based on physics-driven deterministic approaches, utilizing domain knowledge to simulate the physical and chemical processes of vehicle emissions and air pollutants (Shiva Nagendra & Khare, 2004; Smit et al., 2010; Vardoulakis et al., 2003). These models often involve computationally intensive procedures and have limited generalization capabilities. To overcome these shortcomings, researchers have increasingly focused on data-driven methods over the past few years, exploring how machine learning (ML) and deep learning (DL) models can simulate complex spatiotemporal processes. These models can accurately represent nonlinear processes without the need for explicit programming of each physical process, significantly improving computational efficiency. Time series analysis models have been applied to air pollutant forecasting to capture temporal dependencies, such as autoregressive-based models (Chelani & Devotta, 2006) and recurrent neural networks (RNN) (Chen et al., 2019), to predict future road emissions based on historical emission data and meteorological changes. However, these studies often overlook the spatial characteristics of the geographic environment, which play a crucial role in the generation and dispersion of pollutants. Other methods have employed convolutional neural networks (CNN) to capture spatial dependencies (Huang & Kuo, 2018; Xu et al., 2019), with some studies using satellite remote sensing data to estimate surface air pollution (Martin, 2008; van Donkelaar et al., 2006). However, such approaches tend to operate at larger scales, making it difficult to achieve the precision needed for identifying and predicting road pollutants in urban environments. Some methods have also used spatial interpolation techniques to convert sparse monitoring station data into continuous grid-based data for prediction (Chen & Liu, 2012). However, these methods require interpolation over large areas with unknown pollutant levels, which is computationally expensive and prone to errors due to data sparsity.

Therefore, predicting road emissions involves multiple challenges. First, road emissions exhibit complex dependencies across spatial and temporal dimensions, making it difficult for conventional models to simultaneously capture multidimensional interactions. Second,

road emissions are influenced by multiple complex factors such as weather, land use (Zhang & Gong, 2018), and wind-driven dispersion processes (Hettige et al., 2024; Kim et al., 2023), requiring a rich dataset as the foundation for modeling to improve prediction accuracy. Lastly, one of the primary sources of fine-scale road emission data remains ground-based monitoring stations, and how to effectively leverage the rich data generated by sparsely distributed air monitoring stations requires further methodological innovation and exploration.

Recently, graph neural networks (GNN) have been increasingly applied to air pollutant prediction tasks (Zhang et al., 2019). A set of spatially sparse ground monitoring station data can be viewed as typical graph-structured data, with each monitoring station considered a node in the graph, and spatial correlations represented as edges between nodes. These correlations may arise from spatial proximity or meteorological dispersion. Unlike deep learning methods like CNNs, GNNs can learn correlations between sites within the context of complex non-Euclidean spatial information. Additionally, combining GNNs with time series models such as RNNs can effectively handle spatiotemporal data (Jin et al., 2023). For instance, Xu et al. (2020) proposed a hierarchical graph neural network-based air quality forecasting model to simulate the correlation between spatiotemporal dynamic factors and air pollution, predicting across ten major cities in the Yangtze River Delta urban agglomeration. Iskandaryan et al. (2023) used an Attention Temporal Graph Convolutional Network, integrating meteorological and land use data, to perform spatiotemporal NO2 predictions in Madrid. Building on previous research, this study proposes a novel spatiotemporal graph neural network to achieve better road emission prediction accuracy by fully leveraging multi-source spatiotemporal data. The contributions of this study can be summarized as follows:

1. The study effectively integrates the LSTM time series model with graph neural networks based on the constructed station graph structure, fully capturing the complex spatiotemporal dependencies of road emissions.
2. In addition to considering the spatial features such as meteorology and land use adopted by most existing studies, this research also incorporates potential traffic characteristics related to road emissions, including road conditions, structure, and traffic flow, to provide further policy insights.

3. The study incorporates domain knowledge about pollutant dispersion into the modeling process by designing a method to dynamically adjust the spatial correlations between monitoring stations based on wind speed and direction, more accurately capturing spatiotemporal dynamics.

2 Data Preparation

2.1 Road Emission

The road emission dataset is sourced from London Air Quality Network (LAQN), which is a comprehensive system established in 1993 to monitor and analyze air pollution across London and Southeast England. Managed by the Environmental Research Group at Imperial College London, the network consists of around 239 monitoring stations strategically placed in urban areas, of which 78 are near the roadside, where pollution levels are typically higher. These stations collect data on various pollutants, such as nitrogen dioxide (NO2), particulate matter (PM10 and PM2.5), and ozone (O3), at regular intervals, typically every 15 minutes.

The LAQN website provides an API that allows data retrieval based on pollutant type, site category and code, and date range. The dataset used in this paper was collected hourly from roadside stations between January 1, 2023, and December 31, 2023. The study excluded stations that were not fully operational throughout the study period. During the data cleaning process, it was found that most monitoring stations had relatively complete NO2 emission data (more than 50 stations), so NO2 was chosen as the prediction target for this study, which is also one of the main pollutants from traffic emissions (Beevers et al., 2012).

Due to equipment maintenance or other interruptions, data collected by monitoring stations may be incomplete. In the original dataset of this paper, many stations exhibited an amount of missing data. This paper first excludes stations with more than 20% missing data. For the remaining stations, the time series of road emissions is interpolated using linear interpolation. Time interpolation is a technique used in time series data processing to estimate missing values between known time points, thereby filling in the gaps and increasing data resolution. For stations with remaining gaps that cannot be filled by linear interpolation, forward fill is applied to complete the dataset. After two rounds of

interpolation, data from 44 monitoring stations are considered suitable for modeling (shown in Figure 1).

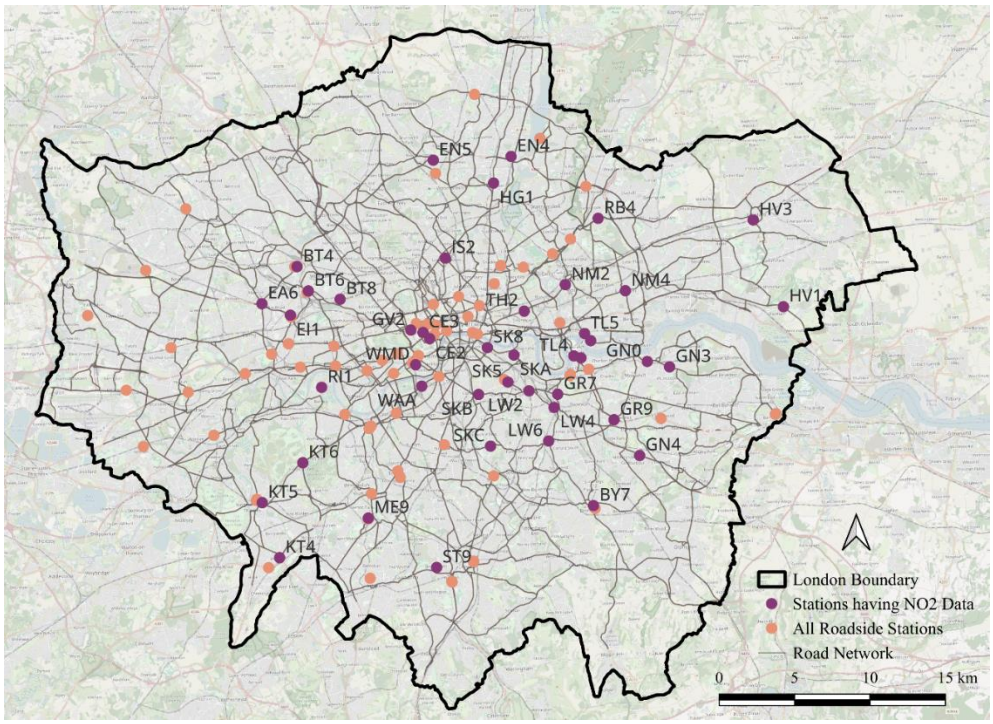

Fig. 1. The distribution of Monitoring stations in London

Figure 2 (a) shows the time series of NO2 concentration at all 44 roadside stations during study period. Some NO2 concentration values were slightly below 0 in the raw data, which is typically caused by minor calibration inaccuracies in the sensors or electronic noise. These values were corrected to 0. Statistical analysis revealed that there were no significant outliers in NO2 concentration during the study period. The maximum value was 235.7 μg/m³ (recorded on 2023-05-04 at 11:00 at station BT4), the minimum was 0 μg/m³, the mean concentration is 26.68 μg/m³ and the standard deviation is 17.78 μg/m³. Figure 2 (b) shows the mean time series of NO2 concentrations across 44 monitoring stations, revealing significant characteristics such as periodicity, proximity, and trend. Periodicity refers to the cyclical fluctuations in NO2 concentrations over an extended period. Proximity indicates that NO2 concentration values are more similar when the time periods are closer together. Trend refers to the increasing or decreasing pattern in NO2 concentrations over a certain period. Therefore, modeling the temporal dependencies in NO2 concentration data is crucial for accurate predictions.

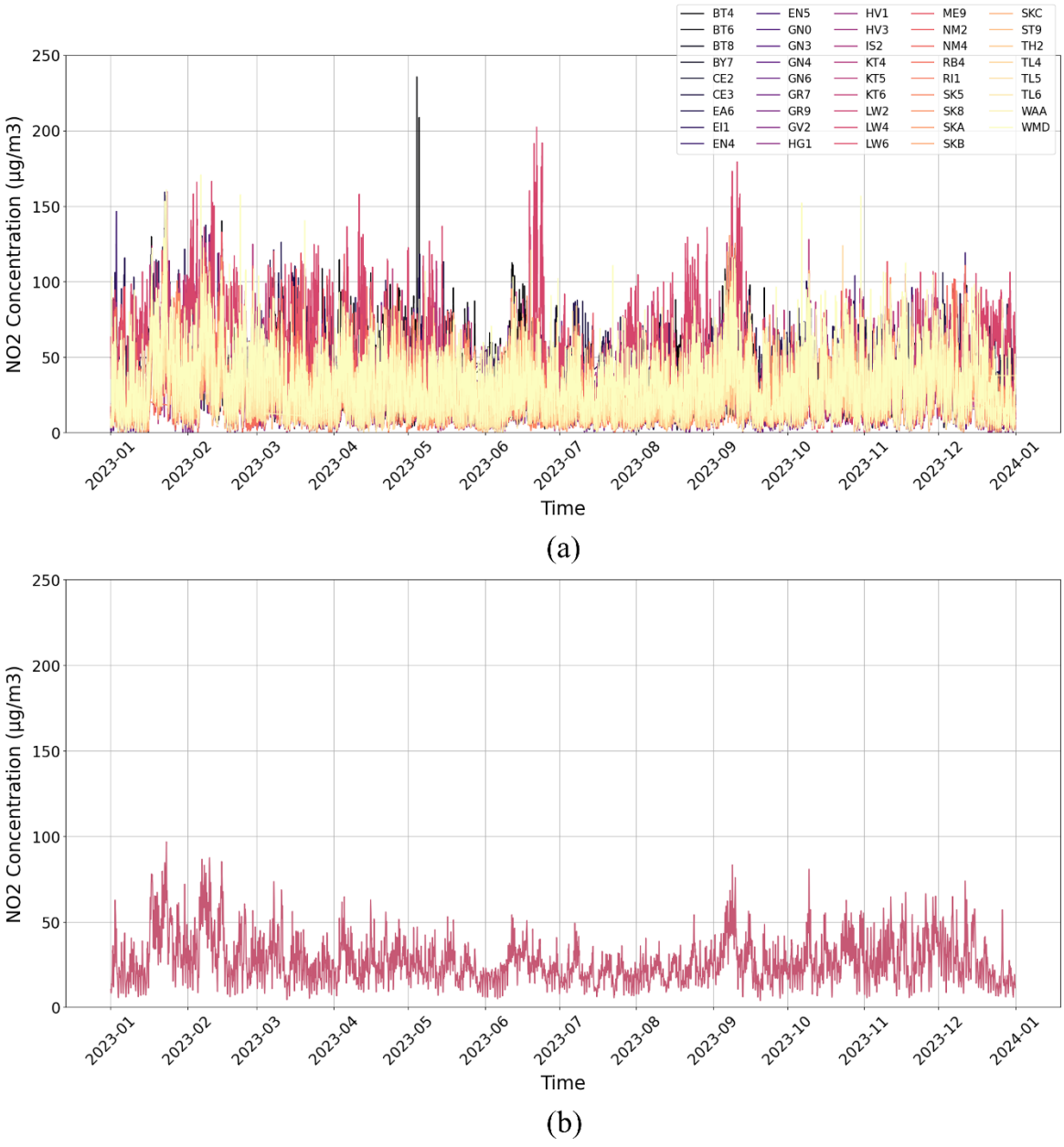

Fig. 2. The time series of NO2 concentration at all stations during study period

Figure 3 presents a box plot of the NO2 time series data for 44 stations, categorized by month and hour, showing discernible trends and patterns in NO2 concentration over different temporal scales. From Figure 3(a), it is evident that NO2 concentration peaks in February, with a median close to 30.80 µg/m³, indicating higher pollution levels. The lower concentrations from March to August may be associated with favorable dispersion conditions during spring and summer, such as higher temperatures and strong ultraviolet

radiation that promote photochemical reactions. NO2 levels in other months are relatively consistent but still exhibit some seasonal fluctuations, with higher concentrations in winter and autumn, likely due to increased emissions during the heating season and poorer dispersion conditions. Figure 3 (b) shows that NO2 concentrations peak between 5 PM and 8 PM, which can be attributed to increased traffic emissions during the evening rush hour. In contrast, concentrations are lower in the early morning hours (midnight to 6 AM) when traffic volume and emission sources are minimal. Overall, the temporal characteristics of NO2 concentration highlighted in the Figure clearly point to the significant impact of traffic activities and seasonal factors on air quality.

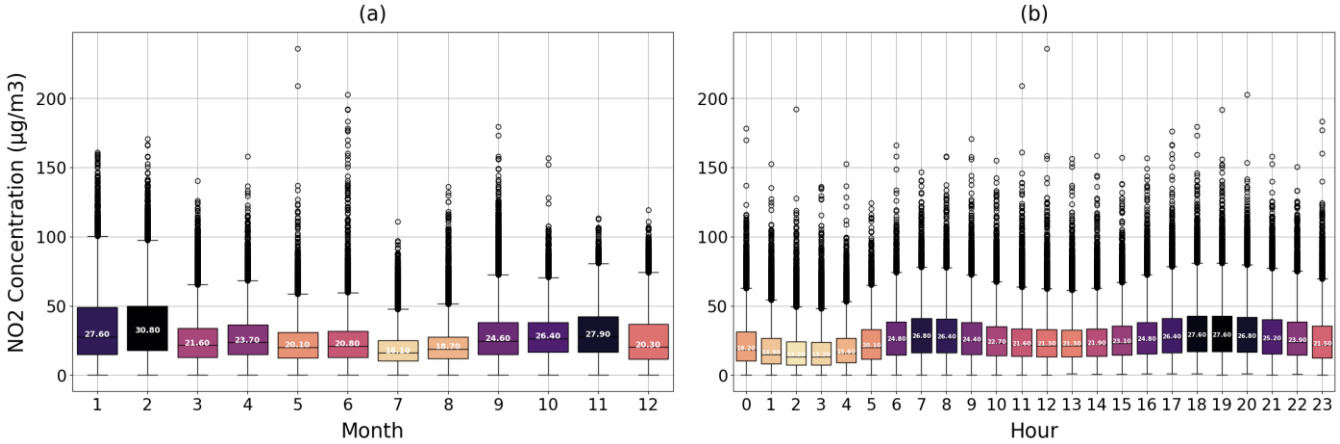

Fig. 3. Box plot of the NO2 concentration time series for stations categorized by month (a) and hour (b)

Another exploratory analysis was conducted to detect the correlation between the NO2 time series at the stations. As shown in Figure 4 (a), the correlations between the NO2 time series of the 44 monitoring stations indicate that most stations have a correlation greater than 0.6, ranging from moderate to strong correlation. Further analysis involved calculating the Haversine distance matrix between the monitoring stations. A scatter plot was generated, with the Haversine distance between stations on the x-axis and the time series correlation on the y-axis, as shown in Figure 4 (b). The plot displays a slightly sloped regression line, indicating a modest negative correlation between distance and time series correlation—that is, as the distance between stations increases, the correlation of their time series tends to decrease slightly. The scatter points around the regression line are somewhat dispersed, suggesting that in addition to spatial proximity, other factors such as local traffic conditions and land use also influence the time series at monitoring

stations. Consequently, more advanced and complex nonlinear modeling techniques are necessary to capture the intricate spatial dependencies between these features.

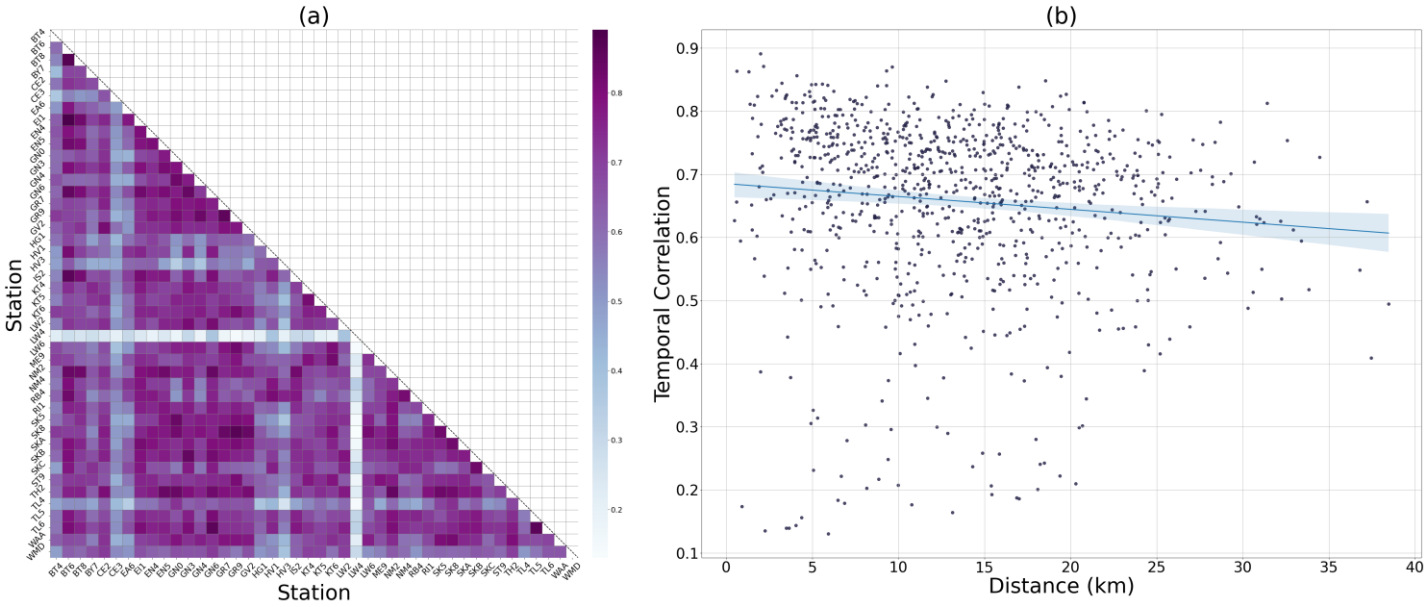

Fig. 4. Correlation matrix of monitoring station time series (a) and regression with spatial distance (b)

2.2 Features Extraction

Previous studies have shown that various local characteristics, such as historical air pollutants, meteorological parameters, and POI-related features, are considered key influencing factors of air pollutants (Wang et al., 2021; Zheng et al., 2013). Additionally, since this paper focuses on roadside emissions, road and traffic characteristics also need to be considered in this study. Based on the spatio-temporal attributes of the features, this paper classifies them into three categories: static, temporal, and global features. The following part will introduce the process of data acquisition and feature extraction for each category.

2.2.1 Static Features

Static features refer to spatial characteristics that remain relatively constant over the study period (i.e., they vary across different monitoring stations but not over time). These features include Land Use, Road Condition, and Traffic features.

(i) Road condition

Road condition contains six features as shown in Table 1 below.

Table 1 Details Of Road Condition Features

Feature name	Description
Road type	different types of roads including highways, major roads, and minor roads, accommodate varying traffic volumes and vehicle types, affecting emission levels
Road length	longer roads may accumulate more emissions, especially in cases of heavy traffic flow
Number of lanes	lanes usually mean higher traffic capacity and thus higher emissions
Maximum speed limit	it influences emissions by affecting vehicle speed and acceleration patterns
One-way road setup	one-way roads might reduce emissions by decreasing vehicle conflicts and wait times
Road gradient	roads with steeper slopes may cause vehicles to accelerate or decelerate, increasing engine workload and thus NO2 emissions
Junction centrality	areas with high centrality (the number of roads connected to a junction) typically experience heavier traffic, leading to increased vehicle emissions

Road condition data is obtained from OpenStreetMap (OSM), a collaborative initiative providing a freely available database for map elements. The OSMnx package (Boeing, 2017) is used to directly retrieve major and minor roads with the attributes mentioned above in London. Road grade is calculated using the OSMnx package with additional elevation data at road intersections from the Google Maps Elevation API. Finally, the features from the nearest road and the centrality of the nearest junctions were assigned to the corresponding station.

(ii) Traffic feature

Traffic feature represents the average daily traffic volume of various vehicle types on major and minor roads in 2023, based on data collected by the UK's Department for Transport from the GB Road Traffic Counts. These traffic counts are distributed across the roads as manual sampling points. For each monitoring station, the traffic volume data of the nearest sampling point was assigned and for each vehicle type was assigned to the corresponding station. Due to differences in engine power and fuel type, the NO2 emissions from different types of vehicles vary significantly (Zeng et al., 2024). For

instance, the vehicles included in this dataset—two-wheelers, cars and taxis, buses and coaches, light goods vehicles (LGVs), and heavy goods vehicles (HGVs)—emit NO2 in varying amounts, generally decreasing in that order.

Although hourly-varying road emissions are more heavily influenced by real-time traffic flow, due to the lack of real-time traffic data during the study period, the 2023 average daily data is used as a substitute for. However, the static traffic features still provide valuable insights into the general characteristics of urban area and road function where a monitoring station is located. For example, areas with high daily truck traffic are typically industrial zones, while areas with high two-wheeler and car traffic are usually commercial or residential zones. Therefore, static traffic features remain a significant factor in the study.

(iii) Land use

In addition to the road conditions and traffic characteristics where the stations are located, the surrounding land use also significantly impacts road emissions. Green spaces can absorb nitrogen oxides (NO2), thereby reducing the concentration of emissions near roads. Conversely, industrial and transport areas near roads may significantly increase pollution levels due to industrial activities and transportation infrastructure. Additionally, high building density can create a "street canyon" to hinder the dispersion of pollutants, leading to increased NO2 concentrations. Therefore, the area ratio of green space, industry and transport, and building density were selected as features.

This paper firstly retrieved raw data from OSM historical data in 2023 using the Ohsome API, and then established a 1km buffer zone around the monitoring stations. The proportion of each land use type and building area relative to the total buffer area was then calculated.

2.2.2 Temporal Features

Temporal features are spatiotemporal characteristics that vary over time and across different stations. These features have a time dependency, meaning that historical time series data influences future values. Temporal features include historical NO2 concentration and Meteorology.

Based on the time series analysis mentioned above, NO2 concentrations at adjacent time periods are more similar. Therefore, the time series of observed historical NO2

concentrations is one of the key influencing features. Here, the length of historical data is selected as 24 hours, in line with most related studies (Ge et al., 2021; Xu et al., 2023; Zhao & Zettsu, 2020).

Meteorological features significantly impact hourly road emissions by affecting the dispersion, chemical reactions, and accumulation of pollutants. Seven meteorological features are selected in total: Wind speed, Pressure, and Precipitation influence the dispersion and accumulation of NO2 concentrations, while Humidity, Temperature, Cloud cover, and UV Index affect the photochemical reactions and transformations of NO2. The dataset is obtained through World Weather Online (WWO) API, which varies across different locations of stations, especially for features like wind speed, humidity, and cloud cover. They are therefore considered as spatiotemporal features rather than global features.

2.2.3 Global Features

Global features are those that affect all monitoring stations equally across the entire study period although they provide temporal context. The features are categorized into two types: Hour and Calendar. The Hour variable is derived from the hour value in the timestamp of road emission data, as the previous analysis indicated significant variations in NO2 concentration at different hours daily. The Calendar feature specifies whether it is a Weekday, Weekend, or Bank Holiday in the UK, based on data obtained from the UK Bank Holidays API. Both of these features influence road emission amounts by affecting traffic activity levels. The reason month was not selected as a feature is that when the annual data is split into training and testing sets, the months covered in each dataset might differ, potentially reducing the model's generalization ability.

3 Methodology

3.1 Problem Formulation

The goal of road emission prediction here is to forecast future NO2 concentration in a given time interval based on spatiotemporal influencing factors gathered from citywide monitoring stations in London. This paper models the changing spatial correlations among different stations by graph structure. Given a directed graph $\mathcal{G} = (\mathbb{V}, \mathbf{E}, \mathbf{K})$, where $\mathbb{V}$ is the set of nodes representing monitoring stations and $|\mathbb{V}| = N$, $\mathbf{E}$ is a set of edges

representing the existing relevance between nodes and **K** is the edge feature matrix indicating the interaction among nodes in terms of wind field (see details in Section 3.3). Let $\mathbf{X}^{static} \in \mathbb{R}^{N\times S}$, $\mathbf{X}^{temporal}_{(t-\tau_{in}+1:t)} \in \mathbb{R}^{N\times P}$, $\mathbf{X}^{global}_{t} \in \mathbb{R}^{N\times Q}$ denote the static node features throughout the entire study period, temporal node features of Past τ_{in} time steps and global features at time step t. S, P, Q represent the number of features corresponding to each type. The length of the historical time window τ_{in} is set to 24 (hour) and the forecasting horizon τ_{out} is set to 12 (hour) in this paper.

The node prediction task aims to learn a function $h(.)$ minimizing prediction error, which can predict NO_2 concentration over the next τ time steps based on proposed node and edge features:

$$\widehat{\mathbf{Y}}_{(t+1:t+\tau_{out})} = h\left(\mathbf{X}^{static},\ \mathbf{X}^{temporal}_{(t-\tau_{in}+1:t)},\ \mathbf{X}^{global}_{t},\ \mathcal{G}\right) \quad (1)$$

3.2 Temporal Sampling

The time-varying dataset in this paper, including road emission data and temporal features, ranges from 2023/01/01 00:00:00 to 2023/12/31 23:00:00, with a total of 8,760 hourly timestamps. As shown in Figure 5 below, we apply a sliding window of length (τ_{in} = 24) (step = 1) for 24-hour historical temporal features and a sliding window of length (t_{in} = 12) (step = 1) for road emission within the 12-hour forecasting horizon. This sliding sampling over the time series results in 8760 - 24 - 11 = 8725 time series samples for both.

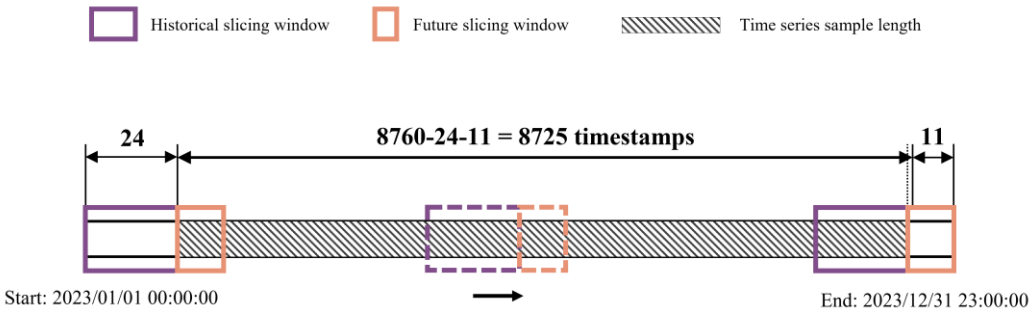

Fig. 5. The illustration of temporal sampling

3.3 Graph Construction

Given the existing and fixed monitoring stations as graph nodes, determining the edges is critical for the construction of a graph. This paper introduces the following two steps to measure dynamic edges between nodes throughout the study period. The first step determines the generation of edges, and the second step further assigns weights to edges.

3.3.1 Spatial Correlation

Due to the fluidity of the air, monitoring stations that are geospatially close may collect correlated results, so the connected edge ϵ_{ij} between node i and j is defined as follows:

$$\epsilon_{ij} = \begin{cases} 1, & 0 < d_{ij} < R \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where d_{ij} denotes the Haversine distance between node i and j , and R is the distance threshold but there is no uniform standard for its value yet. Thus, referencing the relevant literature (Iskandaryan et al., 2023), I adopt an adaptive λ to determine R defined as follows:

$$R = \lambda \max\left(\min\left(d_{i,k}\right)\right), \quad i,k \in V,\ i \neq k,\ \lambda \geq 1 \tag{3}$$

where R is related to the maximum value of the minimum distances from each node to all other nodes and adjusted by a hyper-parameter λ to control the number of edges in the constructed graph (see details in Section 4.2). The smaller the value of λ is, the less the number of edges is. In this way, we can make sure that all nodes in the constructed graph have at least one edge connected to other nodes. R should not be too large to make the graph fully connected.

3.3.2 Wind-driven Dispersal

Existing air pollutant prediction methods construct static graphs based on the geographical distances between monitoring sites (Iskandaryan et al., 2023; Oliveira Santos et al., 2023). However, in the real world, the interaction relationships between monitoring sites are more explicitly manifested in the flow of air pollutants. Advection is a fundamental physical process that describes the movement of air pollutants due to flow

fields (e.g., the bulk movement of air), typically driven by wind patterns. According to relevant literature (Xu et al., 2023), this paper uses the advection coefficient $\vec{\kappa}_{ij}^{t}$ calculated as shown in Equation (4) to simulate the NO2 dispersion trends dynamically adjusted based on the relative positions of stations and real-time wind directions:

$$\vec{\boldsymbol{\kappa}}_{ij}^{t} = \text{relu}\left(\frac{|\vec{\boldsymbol{v}}_{i}^{t}|}{d_{ij}} \cdot \cos\theta_{ij}^{t}\right) \tag{4}$$

Where $\vec{\boldsymbol{v}}_{i}^{t}$ represents the wind speed at node i and time t, which is collected from meteorological data, d_{ij} is the distance between nodes i and j. As the edge weight, the calculation of wind-driven dispersal assumes that the pollutant dispersion between two stations beyond the threshold R is negligible. $relu(.)$ is the ReLU activation function, which applies a nonlinear activation to ensure that all edge attributes are non-negative. θ_{ij} is the angle between the position vector from source node i to destination node j and the wind direction vector at node i and time step t. $\cos\theta_{ij}^{t}$ measures the cosine similarity between two vectors., the smaller the angle, the larger the cosine value, the more likely the air pollutants from node i would be brought to node j by wind. The specific calculation formula is as follows:

$$\overrightarrow{\boldsymbol{\zeta}_{ij}} = \left(x_i - x_j, y_i - y_j\right), \quad \overrightarrow{\boldsymbol{\zeta}_{ji}} = \left(x_j - x_i, y_j - y_i\right) \tag{5}$$

$$\cos\theta_{ij}^{t} = \frac{\overrightarrow{\boldsymbol{\omega}_{i}^{t}} \cdot \overrightarrow{\boldsymbol{\zeta}_{ij}}}{\left|\overrightarrow{\boldsymbol{\omega}_{i}^{t}}\right| \left|\overrightarrow{\boldsymbol{\zeta}_{ij}}\right|}, \quad \cos\theta_{ji}^{t} = \frac{\overrightarrow{\boldsymbol{\omega}_{j}^{t}} \cdot \overrightarrow{\boldsymbol{\zeta}_{ji}}}{\left|\overrightarrow{\boldsymbol{\omega}_{j}^{t}}\right| \left|\overrightarrow{\boldsymbol{\zeta}_{ji}}\right|} \tag{6}$$

where $\cdot$ denotes inner product operation, $\overrightarrow{\boldsymbol{\zeta}_{ij}}$ and $\overrightarrow{\boldsymbol{\zeta}_{ji}}$ denote the position vectors, calculated by the difference between the projected coordinates x_i, y_i of node i and x_j, y_j of node j. $\vec{\boldsymbol{\omega}}_{i}^{t}$ and $\vec{\boldsymbol{\omega}}_{j}^{t}$ denote the wind direction vector at node i and j at time t, also obtained from meteorological data and divided into 16 directions. They are encoded according to the encoding rules from (Xu et al., n.d.) shown in Table 2.

Table 2 The Encoding Rules Of Wind Direction

Direction	Vector
East (E)	[1, 0]
East-Northeast (ENE)	[cos22.5°, sin22.5°]
Northeast (NE)	[cos45°, sin45°]
North-Northeast (NNE)	[cos67.5°, sin67.5°]
North (N)	[0, 1]
North-Northwest (NNW)	[cos112.5°, sin112.5°]
Northwest (NW)	[cos135°, sin135°]
West-Northwest (WNW)	[cos157.5°, sin157.5°]
West (W)	[−1, 0]
West-Southwest (WSW)	[cos202.5°, sin202.5°]
Southwest (SW)	[cos225°, sin225°]
South-Southwest (SSW)	[cos247.5°, sin247.5°]
South (S)	[0, −1]
South-Southeast (SSE)	[cos292.5°, sin292.5°]
Southeast (SE)	[cos315°, sin315°]
East-Southeast (ESE)	[cos337.5°, sin337.5°]

3.4 Proposed Model

As previously mentioned, the air pollution concentration at a specific location and time is influenced by neighboring areas in space and proximate moments in time. Therefore, road emission prediction can be regarded as a spatiotemporal dependency modeling problem. The proposed deep learning model in this paper is divided into two components: spatial dependency modeling and temporal dependency modeling, with a detailed explanation provided afterward on how these components are integrated to obtain the final prediction results.

3.4.1 Spatial Dependency Modeling

Traditional neural networks, such as Convolutional Neural Networks (CNNs), are designed to process data that lie on a regular grid, such as images, sequences, and raster data in the field of geosciences, where the data points have a natural spatial relationship that can be captured using Euclidean geometry. However, in this study, the spatial relationships between NO_2 concentrations at monitoring stations are irregular and

complex, especially in the context of related domain knowledge. Graph Neural Networks (GNNs) are specifically designed to handle this non-Euclidean structure by learning representations of nodes, edges, and entire graphs, while respecting the underlying graph topology. This allows GNNs to capture the relational information between nodes in a way that traditional Euclidean-based methods cannot.

Nodes with spatial dependencies in GNNs are referred to as neighbors. GNNs aggregate the feature information from neighboring nodes onto the target node and update its feature representation through nonlinear transformations to improve the accuracy of node predictions. This entire process is known as message passing. The non-Euclidean spatial relationships between nodes this paper established previously, which are based on prior knowledge of geographic proximity and wind-driven dispersal, determine the way messages such as historical road emissions, meteorological, and geographical features are passing between nodes. Here, we further employ the Graph Attention Network (GAT), which automatically captures the importance of one node to another by calculating attention coefficients, reflecting the strength of the relationship between the two nodes, and applying this in the message passing process. The following formula can calculate the attention coefficient e_{ij} between neighboring node i and target node j:

$$\mathbf{e}_{ij} = \text{LeakyReLU}\left(\mathbf{a}^T\left[W\mathbf{h}_i \parallel W\mathbf{h}_j\right]\right) \tag{7}$$

Where $\mathbf{a}^T$ is a learnable parameter vector that is used to compute the attention scores, $.^T$ denotes transposition. $\mathbf{h}_i$ and $\mathbf{h}_j$ are the node representations initialized by the feature vectors of nodes i and j and continually updated by message passing further. W is a learnable shared weight matrix giving node representations a linear transformation and $\parallel$ represents the concatenation operation. The concatenated feature vectors will be done an inner product with $\mathbf{a}^T$. LeakyReLU is a nonlinearity function that allows a small, non-zero gradient for negative inputs.

To make the attention coefficient comparable between all nodes, the softmax function is used to normalize the attention coefficient as follows:

$$\boldsymbol{\alpha}_{ij} = \text{softmax}\left(\mathbf{e}_{ij}\right) = \frac{\exp\left(\mathbf{e}_{ij}\right)}{\sum_{k\in N_i}\exp\left(\mathbf{e}_{ik}\right)} \tag{8}$$

where N_i represents all neighboring nodes of node i (including i) in the constructed graph. From the mathematical principle reflected in the formula, it is evident that when the neighboring nodes have similar features and the edge attributes have larger values, the concatenated vector, after passing through the nonlinear activation function, can output a higher attention score. If the model learns through training that aggregating more information from this neighboring node can improve the accuracy of predictions at the target node, it will adjust the parameters $\mathbf{a}^T$ and W to further increase the weight of that neighboring node. After calculating the weight of each node's neighboring nodes, the output h_i' of the GAT layer can be obtained by aggregating the weighted information of the neighboring nodes:

$$\mathbf{h}_i' = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\, W\left[\mathbf{h}_i \parallel \mathbf{h}_j \parallel \vec{\boldsymbol{\kappa}}_{ij}\right]\right) \tag{9}$$

where σ represents the activation function, W is still a learnable weight matrix. The features of node i itself are also spliced into the information aggregation to ensure that nodes retain their own features during the update process. Edge attribute, the wind-driven advection coefficient $\vec{\boldsymbol{\kappa}}_{ij}$ calculated above is concatenated additionally with the node features in this paper, contributing to the information aggregation. Meanwhile, GAT can distinguish directed edge attributes, namely the wind field information past from node i pointing to node j and from node j pointing to node i is not the same, which aligns with the real-world scenario in this paper.

In addition to a separate attention mechanism, multi-head attention can ensure the stability of the attention mechanism and allow the model to have the to learn relevant information from different subspaces. The specific process is calculated as follows:

$$\mathbf{h'}_i = \Vert_{k=1}^{K}\left(\sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k\, W^k \mathbf{h}_j\right)\right), \quad \mathbf{h'}_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k\, W^k \mathbf{h}_j\right) \tag{10}$$

Where K represents the number of attention layers, indicating the attention mechanism will be executed in parallel K times, with each execution referred to as a "head". Each head has its own independent W^k and α_{ij}^k. The formula on the left and the formula on the

right respectively represent the concatenation or averaging of the outputs from each head, with the final aggregated result passing through an activation function σ to obtain the updated feature representation h_i' of node i.

In this study, two layers of GAT were employed. The first layer was configured with 8 attention heads, each performing independent graph convolution operations on the input features. The outputs were then concatenated to increase the feature representation's dimensionality. In contrast, the second layer aggregated the multi-head information by averaging, thereby reducing the feature dimensions, which served as the final output of the spatial dependency component.

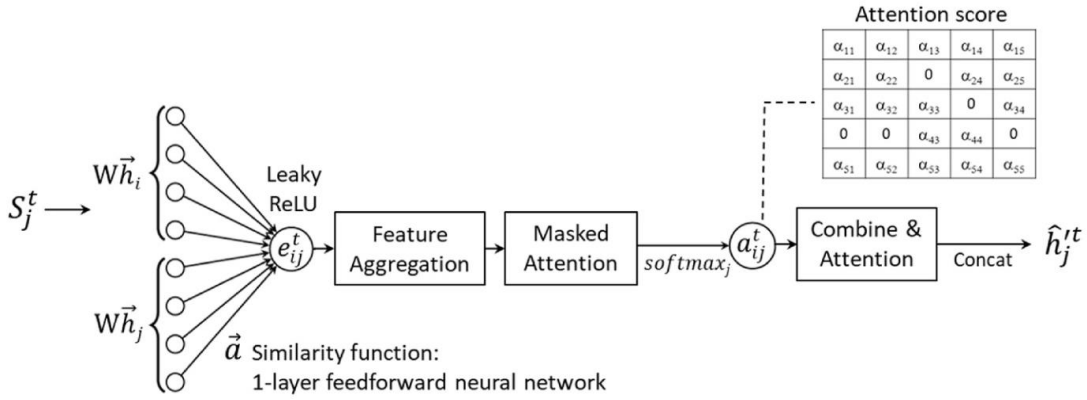

Fig. 6. GAT mechanism (Veličković et al., 2017)

3.4.2 Temporal Dependency Modeling

Long-term dependency refers to the phenomenon in sequential data (e.g., temporal data) where historical information has a significant impact on the current or future output. In such cases, the model needs to "remember" key information from the past in order to utilize it for making predictions or decisions when necessary. Introduced by Hochreiter & Schmidhuber (1997), the proposed model applies LSTM (Long Short-Term Memory) to model the temporal dependencies, which is a type of recurrent neural network (RNN) that incorporates a memory cell and gating mechanisms with three gates (input gate, forget gate, and output gate) to effectively preserve and transmits important historical information throughout the sequence, enabling the network to retain long-term dependencies and make better predictions.

An important concept in LSTM is the Cell State, which is a vector that runs through the entire sequence, serving as the "memory" of the LSTM. The Cell State is updated at each

time step through the forget gate and input gate, the network can selectively retain important information while discarding irrelevant information. This allows the Cell State to preserve crucial information over long sequences, aiding the model in capturing long-term dependencies. Given a sequence data $\mathbf{X} = \{\mathbf{x}_{t-\tau_{\text{in}}+1}, \mathbf{x}_{t-\tau_{\text{in}}+2}, \dots, \mathbf{x}_t\}$ with $\mathbf{x}$, t and τ_{in} denoting temporal feature vector, time step and historical time window respectively**,** LSTM sequentially processes each time step in this sequence, with the total number of time steps being τ_{in}. The detailed implementation of LSTM is as follows, where W_f , W_i, W_C and W_o are weight matrix for corresponding gates, b_f, b_i, b_C and b_o are the bias of gates, independently learned by the model during training:

Forget Gate: Decides which part of the previous cell state C_{t-1} should be kept.

$$f_t = \sigma\left(W_f \cdot [\mathbf{h}_{t-1} \parallel \mathbf{x}_t] + b_f\right) \tag{11}$$

Where f_t is the forget gate output, h_{t-1} is the previous hidden state which is the output of last time step t-1 in the same LSTM layer and its initial value is usually a vector of zeros, indicating that there is no historical information at the initial moment. $\mathbf{x}_t$ is the inputting feature vector at time t, and σ is the sigmoid activation function.

Input Gate: Determines how much new information should be added to the cell state.

$$i_t = \sigma(W_i \cdot [\mathbf{h}_{t-1} \parallel \mathbf{x}_t] + b_i) \tag{12}$$

$$\tilde{C}_t = \tanh(W_C \cdot [\mathbf{h}_{t-1} \parallel \mathbf{x}_t] + b_C) \tag{13}$$

Where i_t is the input gate output, and $\tilde{C}_t$ is the candidate cell state. The $\tanh$ function compresses the result into the range [-1, 1], ensuring that the candidate cell state can represent a wide range of positive and negative information.

Cell State Update: Combines the previous cell state C_{t-1} with the new information to update the current cell state.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{14}$$

Where $\odot$ denotes Hadamard product operation (the element-wise multiplication), combining the forget gate's output with the previous cell state and the input gate's output with the candidate cell state to form the new cell state C_t.

Output Gate: Decides what part of the cell state should be output.

$$o_t = \sigma(W_o \cdot [\mathbf{h}_{t-1} \parallel \mathbf{x}_t] + b_o) \tag{15}$$

$$\mathbf{h}_t = o_t \odot \tanh(C_t) \tag{16}$$

Where o_t is the output gate's output, and h_t is the hidden state, as the output of the LSTM cell at time step t. As the time steps processed increase, the hidden state of the LSTM gradually integrates important information from the input sequence. Therefore, the hidden state at the last time step contains the key information of the entire input sequence, serving as a condensed representation of the input sequence.

To capture temporal dependencies, the proposed paper adopts an encoder-decoder architecture. Both the encoder and decoder consist of several layers of LSTM. The encoder LSTM takes the sequence data $\mathbf{X}_i$ as input and the final hidden state at time step t is obtained after the aforementioned information processing, which represents the cumulative information of the input sequence and is used to initialize the hidden state of the decoder LSTM. During the forecasting horizon τ_{out}, the decoder sequentially processes each future time step, with the resulting hidden state undergoing a nonlinear transformation to output the forecasting result for the current time step. At time step t+1, the decoder uses the known ground truth value at time step t along with the temporal features of time step t+1 as input. For the subsequent τ_{out}-1 time steps, the decoder uses the prediction value from the previous time step and the current time step's temporal features as input. In other words, each time step's output depends on the output of the previous step, ultimately iterating to obtain the predictions for the future τ_{out} time steps.

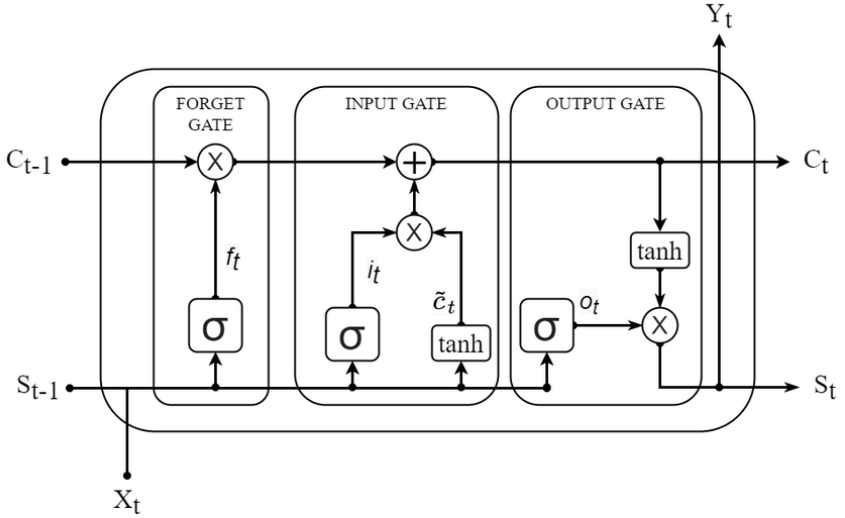

Fig. 7. LSTM mechanism (Hochreiter & Schmidhuber, 1997)

3.4.3 Spatiotemporal Fusion

Fusion models are commonly used in complex tasks such as spatiotemporal prediction, which combine different types of models to handle different dimensions of feature inputs, with the goal of leveraging their respective strengths to improve overall prediction performance. This paper's proposed model can be seen as a stacked structure consisting of a spatial modeling component GAT and a temporal modeling component LSTM. Therefore, it is named REGAL (Road Emission GAT-LSTM). Currently we have the feature matrices $\mathbf{X}^{static}$ with dimensions (*num_nodes*, *num_feats_s*), $\mathbf{X}^{temporal}$ with dimensions (*num_timestamps*, *num_nodes*, τ_{in} , *num_feats_t*) and $\mathbf{X}^{global}$ with dimensions (*num_timestamps*, *num_feats_g*), where *num_nodes* represents the number of station nodes (44), *num_timestamps* represents the number of targeted prediction time points (8725), and *num_feats_s*, *num_feats_t*, *num_feats_g* represents the number of static features (19), temporal features (8) and global features (4) respectively.

The whole process of the proposed model is shown in the following equations. Since the values in the node feature matrices $\mathbf{X}^{static}$ and $\mathbf{X}^{temporal}$ vary with the locations of the stations and station nodes with similar spatio-temporal features often observe similar road emission values, the fusion model firstly implements GAT to perform weighted information aggregation of the concatenated static and temporal features from neighboring nodes to each target node as Equation (17), leveraging the topological relationships and edge attributes contained in $\mathcal{G}$. Here, we treat the node features at different timestamps as different samples entering the GAT. Although the spatial topology of the same node remains identical across different timestamps, the node features and edge attributes vary between samples. The output of GAT layers $\mathbf{X}^{G}$ has dimensions (*num_timestamps*num_nodes** τ_{in} , *num_feats_st*) where *num_feats_st* denotes the dimension of new feature representations that integrate spatial contextual information. This output is then reshaped to the dimensions (*num_timestamps*num_nodes*, τ_{in} , *num_feats_st*) and concatenated with the time-varying global features as input to LSTM encoder component seen in Equation (18). As previously mentioned, the LSTM encoder learns the information of each historical time step in the second dimension of input data and accumulates it into the hidden state h_0 at the final time step, which is used as the initial hidden state for the LSTM decoder.

Additionally, at each future time step, the input data also includes $\widehat{\mathbf{Y}}_t$ and $\mathbf{X}_{t+1}^{future}$ ($t = 1,2,\dots,\tau_{out}$). $\widehat{\mathbf{Y}}_t$ represents the predicted value from the previous time step, and $\mathbf{X}_{t+1}^{future}$represents the known features at the current time step, including calendar and meteorological characteristics. Finally, a multilayer perceptron (MLP) is used to perform a nonlinear transformation on the decoder output to obtain the final emission predictions for the future τ_{out} time steps.

$$\mathbf{X}^G = \mathrm{GAT}\left(\left[\mathbf{X}^{static} \parallel \mathbf{X}^{temporal}\right], \mathcal{G}\right) \tag{17}$$

$$\mathbf{h}_0 = \mathrm{LSTM}([\mathbf{X}^G \parallel \mathbf{X}^{global}]) \tag{18}$$

$$\widehat{\mathbf{Y}}_{t+1}, \mathbf{h}_{t+1} = \mathrm{FCN}\left(\mathrm{LSTM}\left(\mathbf{h}_t, \left[\widehat{\mathbf{Y}}_t \parallel \mathbf{X}_{t+1}^{future}\right]\right)\right), \quad t = 1,2,\dots,\tau_{out} \tag{19}$$

4 Experiments

4.1 Training and Evaluation

4.1.1 Training

The road emission forecasting task aims to get the forecasted NO2 concentration of the monitoring stations for the next τ_{out} time slots. Model REGAL is trained through backpropagation by minimizing the average of Mean Square Error (MSE) with L2 loss between predicted values and ground truth values of τ_{out} time steps and 44 nodes. The loss function in the training phase is as follows:

$$\mathcal{L}(\theta) = \frac{1}{\tau_{out}} \sum_{k=1}^{\tau_{out}} \left(\frac{1}{M} \sum_{i=1}^{M} \left(\hat{Y}_i^{t+k} - Y_i^{t+k} \right)^2 \right) + \gamma \mathcal{L}_{reg} \tag{20}$$

Where *t* represents the first prediction time step in each time series, τ_{out} is forecasting horizon, *M* represents the total number of samples (*num_timestampes* × *num_nodes*). $\hat{Y}_i^{t+k}$ and Y_i^{t+k} represent the predicted value and the ground truth of the sample *i* at time step $t+k$ respectively. $\mathcal{L}_{reg}$ is regularization term for all learnable parameters θ in model to avoid over-fitting and γ is the weighted decay controlling the weight of $\mathcal{L}_{reg}$, considered as a hyperparameter in this study.

4.4.2 Evaluation

To evaluate the performance of the models (REGAL model and other baselines, as seen in the next section), this paper chooses the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), both widely used in regression tasks as evaluation metrics. RMSE measures the geometric difference between the estimated and actual values, and is sensitive to significant errors while MAE measures the average magnitude of the errors. Additionally, Pearson's Correlation Coefficient (PCC) is introduced to measure the linear correlation between two variables. In the experiments, forecasting horizon τ_{out} is divided into three intervals: 1-4 h, 5-8 h, 9-12 h and the mean value of the metrics for each interval is reported. These metrics at each time step are defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(\hat{Y}_i - Y_i\right)^2} \tag{21}$$

$$\text{MAE} = \frac{1}{M}\sum_{i=1}^{M}\left|\hat{Y}_i - Y_i\right| \tag{22}$$

$$R = \frac{\sum_{i=1}^{M}\left(\hat{Y}_i - \overline{\hat{Y}}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_{i=1}^{M}\left(\hat{Y}_i - \overline{\hat{Y}}\right)^2 \sum_{i=1}^{M}\left(Y_i - \overline{Y}\right)^2}} \tag{23}$$

Where $\hat{Y}_i$ and Y_i represent the predicted value and the ground truth of the sample i respectively, $\overline{Y}$and $\overline{\hat{Y}}$ represent the average values of all $\hat{Y}_i$ and Y_i , M represents the total number of samples. Regarding RMSE and MAE, the lower the value is, the better the prediction will be. In terms of PCC, the values range from −1 to +1, where −1 means a perfectly linear negative correlation, +1 means a perfectly linear absolute correlation, and 0 means no correlation.

4.2 Experimental Settings

In this section, we present the experiments to evaluate model REGAL. The detailed descriptions of the dataset and experimental settings are provided in subsection (1). How distance weight λ affects the performance of REGAL is investigated in subsection (2).

The baselines used for performance comparison with the model proposed in this study are introduced in subsection 4.2.3.

4.2.1 Implementation Details

In the feature dataset, categorical variables were encoded using one-hot encoding, while continuous variables were standardized using z-score normalization $X' = \frac{X-\mu}{\sigma}$ in the experiments. To avoid data leakage, the standardization process is conducted after the dataset has been split, ensuring the independence of the training, validation, and test sets. When comparing models, the predicted results and evaluation metrics are then rescaled back to their original values.

Since the prediction target of this study is time series, the dataset needs to be split chronologically in a ratio of 0.7:0.1:0.2 to create non-overlapping training, validation, and test data. Therefore, the training dataset spans from 2023-01-01 00:00:00 to 2023-08-17 11:00:00, the validation dataset spans from 2023-08-17 12:00:00 to 2023-09-23 23:00:00, and the test dataset spans from 2023-09-24 00:00:00 to 2023-12-31 23:00:00.

The experiments are conducted on LSE's Fabian High Performance Computing (HPC) service, utilizing a powerful setup with 32-64 CPU cores and 384-512GB of memory. This cloud-based infrastructure, designed for scalability and security, supports compute-intensive tasks and large dataset processing, making it perfectly suited for the computational demands of this study. The proposed model is implemented in PyTorch 2.2.1 deep learning framework of the server and constructs GNNs with PyTorch Geometric library.

In the experiments, the Adam optimizer was used in training phase with exponential decay (Han et al., 2023). All deep learning models were trained for up to 100 epochs with an early stopping rule of 20 steps, and each model was run 5 times without fixed random seeds to avoid contingency. The experiment will start recording the best validation average RMSE over a 12-hour forecast horizon after 70% of the epochs. If this average RMSE improved compared to the previously recorded best value, we updated it and recorded the corresponding hyperparameter combination along with the RMSE, MAE, and PCC metrics on the test set. If, before reaching 70% of the epochs, the model went through 20 rounds without improvement in validation RMSE, training was also halted to

prevent overfitting, and the current best validation RMSE and corresponding test metrics were reported. Finally, we calculated the average test RMSE corresponding to the best validation RMSE across the 5 runs. This approach ensured the robustness and generalization of hyperparameter selection.

4.2.2 Parameterized Graph Test

The propagation of nitrogen oxides in urban environments is specifically influenced by various factors (e.g., topography and meteorological conditions). Statistical analysis shows that when distance weight λ is in the range of [1, 1.5], the average distance between nodes in the constructed graph is within 5 km, which can be considered a reasonable distance for NO_2 propagation in urban context. Moreover, from a graph-based perspective, almost all the nodes in the constructed graphs are in the largest connected component, meaning there is a path between any two nodes, regardless of how many other nodes and edges the path traverses. This ensures that the information within the graph can be maximally utilized during message passing in GNNs. However, λ significantly influences the number of edges, as increasing it from 1.0 to 1.5 nearly doubles the number of edges (from 312 to 556). Therefore, to more accurately capture the existing relevance between nodes and improve model prediction performance, we first train and evaluate the model's prediction performance for different λ values within the [1, 1.5] range with a step size of 0.1 to determine the optimal λ.

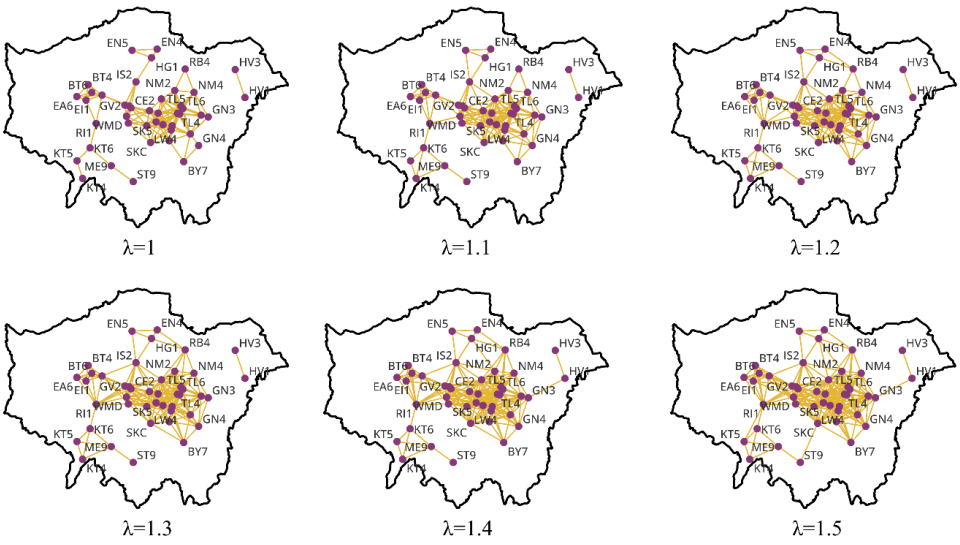

Fig. 8. Station Graphs with different λ

Experiments revealed that the REGAL model performed best when λ equals 1.2. As λ increases, RMSE generally decreases at first and then increases. This may be because when λ is too small, the monitoring stations do not obtain sufficient information from neighboring stations while when λ is too large, the local information at the monitoring stations becomes overly diluted during the message passing process.

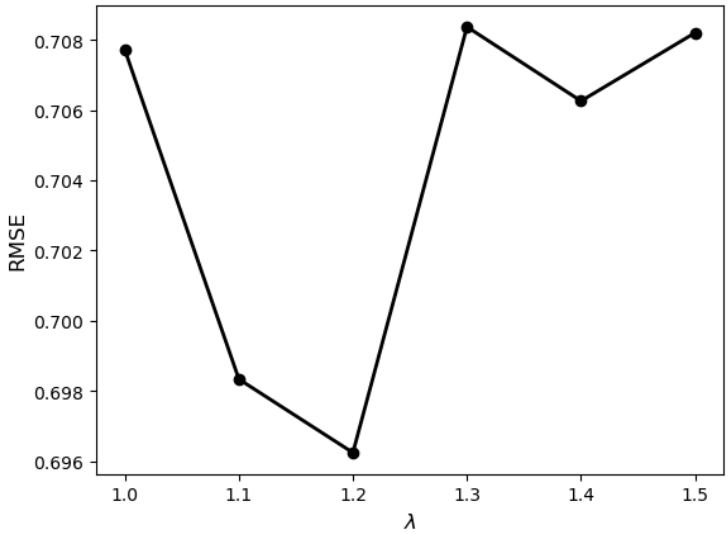

Fig. 9. The evaluation results of distance weight λ

4.2.3 Baseline models

In the prediction experiment, the following models are used as baselines compared with the proposed REGAL model, including one statistical model and four deep learning models.

VAR: Vector Autoregression Regression is a multivariate time series model used to capture the linear interrelationships between multiple time series variables and predict the independent variable for each time series (Ouyang et al., 2021). This paper selects NO2 concentration, meteorology, and calendar time series as inputs. The model is trained on the training set to select the optimal lag order and is then used to perform multi-step forecasting (set to 12) on the NO2 concentration time series in the validation and test sets.

MLP: Multi-layer perceptron is the basic neural network method for regression, which generally consists of an input layer, a hidden layer, and an output layer, with nodes fully connected between layers (Wang et al., 2022). The historical time step dimension of the temporal features is merged into the feature dimension and concatenated with the static

features and global features as the input for the MLP. Set output dimension to 12 as the final prediction outcomes.

GCN: Graph Convolutional Network (GCN) is one of the basic GNN models used for spatial dependency modeling (Zhang et al., 2019). The static features are concatenated with the temporal features, where the historical time step dimension has been merged into the feature dimension, to serve as the input for the GCN. The output, after spatial information aggregation, is then concatenated with the global features and passed through a fully connected layer to produce a 12-dimensional prediction.

GRU: Gated Recurrent Unit is another variant of the recurrent neural network besides LSTM, which is also used to model the temporal features data but has a simpler architecture (Dey and Salem, 2017). Here, one GRU is used as an encoder to encode historical observations, and another GRU is employed to make predictions recursively. Static features and global features are reshaped and directly concatenated with the temporal features as the input for the GRU, without any spatial information aggregation.

HighAir: HighAir is a state-of-the-art spatiotemporal model proposed by Xu et al. (2020). It utilizes a graph neural network with a custom message passing mechanism and LSTM encoder-decoder to model spatial and temporal dependencies, respectively. Unlike this study, the model does not employ a graph attention mechanism and does not consider global features in temporal modeling. The model is based on a hierarchical graph structure, which has been simplified in this study to better align with the data composition used here.

For the deep learning models, we globally tuned the learning rate, L2 regularization's weight decay, and dropout probability. Local settings focused on optimizing the hidden layer dimensions and the number of layers for each model. As previously mentioned, the mean and standard deviation of the RMSE, MAE, and PCC evaluation metrics were recorded after training each baseline model five times. The results of hyperparameter tuning are shown in the Table 3. The bold font denotes the best setting of each hyperparameter.

Table 3 Details Of Grid Search In Each Method

Models	Searching range of hyperparameters	
	Global	Local
VAR	Learning rate: [0.0005, **0.001**, 0.01, 0.1], L2 Regularization term: [0, 0.0005, **0.001**, 0.01], Dropout probability: [0.4, **0.5**, 0.6]	Lag Order: [**5**, 10, 15]
MLP		Linear hidden size: [32, 64, **128**]
GCN		GCN hidden size: [**32**, 64, 128], Number of GCN layers: [1, **2**, 3]
GRU		GRU hidden size: [32, **64**, 128], Number of GRU layers: [1, **2**, 3]
HighAir		GNN hidden size: [32, 64, **128**], LSTM hidden size: [32, **64**, 128], Number of LSTM layers: [1, **2**, 3]
REGAL		GAT hidden size: [32, **64**, 128], Number of GAT layers: [1, **2**, 3], LSTM hidden size: [32, 64, **128**], Number of LSTM layers: [1, **2**, 3]

4.3 Results and Comparison

In this section, proposed REGAL model is compared with other baselines in terms of evaluation metrics and prediction curves in the first subsection. The effectiveness of features used and model components would be discussed in the second subsection, respectively.

4.3.1 Predictive Performance

As shown in the Table 4, the best results are highlighted in bold. The values to the left of the ± symbol represent the mean RMSE within the time interval, while the values to the right represent the standard deviation, which are calculated based on five training times. To more intuitively understand the model's prediction accuracy, the initially standardized metrics are multiplied by the standard deviation of the data to return to the original units. Since VAR employs deterministic statistical methods during estimation, the generated prediction results are deterministic given the same data and parameters, resulting in a standard deviation of 0.

The proposed REGAL model achieves the best performance across all time intervals and evaluation metrics. The correlation between the predicted time series of REGAL and the ground truth reaches as high as 84%, and the RMSE is as low as 9.46, given the standard deviation of approximately 17.62 in the test dataset of the original NO2 concentration, demonstrating the superiority of REGAL. Compared to the second-best model, HighAir, the REGAL model reduces the average MAE and RMSE by 3.4% and 5.0%, respectively in the 1-4 h interval, while increasing PCC by 0.01. Both models utilize edge attributes derived from wind speed and direction between urban nodes while the primary difference is that this study employs a graph attention mechanism, indicating that graph attention networks effectively model the dynamic connections between monitoring nodes, particularly when meteorological knowledge is integrated.

Among individual models, GRU exhibits the best performance, validating the effectiveness of modeling temporal dependencies for time series data prediction. Furthermore, the GCN prediction model, which models spatial dependencies using a graph structure, performs slightly worse than GRU but outperforms MLP and VAR models. These results suggest that modeling spatiotemporal dependencies is crucial for road emission prediction. MLP shows the weakest performance among all models, as it lacks the capability to model advanced representations of spatiotemporal data. Additionally, deep learning methods outperform the classical VAR method, showcasing their ability to learn complex relationships, unlike VAR, which only models linear correlations based on time series analysis, making it less effective in predicting spatiotemporal road emissions.

In terms of time span, for all models, the predictive metrics are best in the 1-4 h interval, with performance declining as the prediction time span increases. This is because road emissions are influenced by multiple factors, and the interactions between these factors are highly complex, making it increasingly difficult to capture spatiotemporal patterns over longer prediction periods. Nevertheless, the REGAL model successfully maintains its high prediction accuracy advantage in long-term prediction tasks, with the prediction error difference between REGAL and other models widening further; for instance, the RMSE reduction relative to the HighAir model increases from 3.4% to 5.7% and then to 7.1%. This advantage can be attributed to the edge attributes constructed from wind

speed and direction in the proposed model, effectively capturing complex spatiotemporal patterns and relationships between monitoring stations. Moreover, the performance gap between GRU and spatiotemporal fusion models gradually widens in long-term predictions, as GRU fails to effectively explore spatial correlations, and its receptive field is limited to the temporal domain.

Table 4 Performance Metrics In Terms Of Time Granularities

Model		RMSE	MAE	PCC
VAR	1-4h	14.31±0.00	12.56±0.00	0.70±0.00
	5-8h	15.46±0.00	13.89±0.00	0.54±0.00
	9-12h	24.61±0.00	23.58±0.00	0.46±0.00
MLP	1-4h	12.46±0.12	9.42±0.12	0.71±0.01
	5-8h	14.51±0.2	11.06±0.16	0.58±0.01
	9-12h	15.64±0.2	12.04±0.13	0.49±0.01
GCN	1-4h	10.16±0.05	7.44±0.04	0.82±0.0
	5-8h	13.03±0.25	9.7±0.2	0.68±0.01
	9-12h	14.35±0.53	10.78±0.42	0.59±0.03
GRU	1-4h	9.87±0.18	7.05±0.19	0.83±0.01
	5-8h	12.85±0.18	9.46±0.19	0.7±0.01
	9-12h	13.43±0.18	10.02±0.19	0.67±0.01
HighAir	1-4h	9.8±0.12	7.1±0.06	0.83±0.01
	5-8h	12.74±0.06	9.45±0.03	0.71±0.0
	9-12h	13.59±0.39	10.08±0.22	0.67±0.01
REGAL	1-4h	**9.46±0.01**	**6.75±0.05**	**0.84±0.0**
	5-8h	**12.01±0.19**	**8.92±0.22**	**0.74±0.01**
	9-12h	**12.63±0.2**	**9.44±0.21**	**0.71±0.01**

This paper visualizes the prediction fit curves and the actual time series for two typical monitoring stations during the period from October 19, 2023, to October 25, 2023 within the test dataset to further examine the detailed differences in prediction performance among the models. The first monitoring station happened to have missing data during this period, which was filled in using linear interpolation, resulting in a straight line. As seen in Figure 10 (a), despite some performance differences, all models were able to learn the intrinsic nonlinear variations in the time series from the complete dataset, thereby fitting a relatively actual NO2 concentration curve. However, Figure 10 (b) reveals that the proposed REGAL model closely follows the actual data in both the upward and downward trends, though there is a slight increase in prediction deviation when the true

NO2 concentration is at a low point. When the actual NO2 concentration reaches a peak, all models except REGAL tend to overestimate the time series values, which may be related to the increased complexity of influencing factors when road emissions are at higher pollution levels.

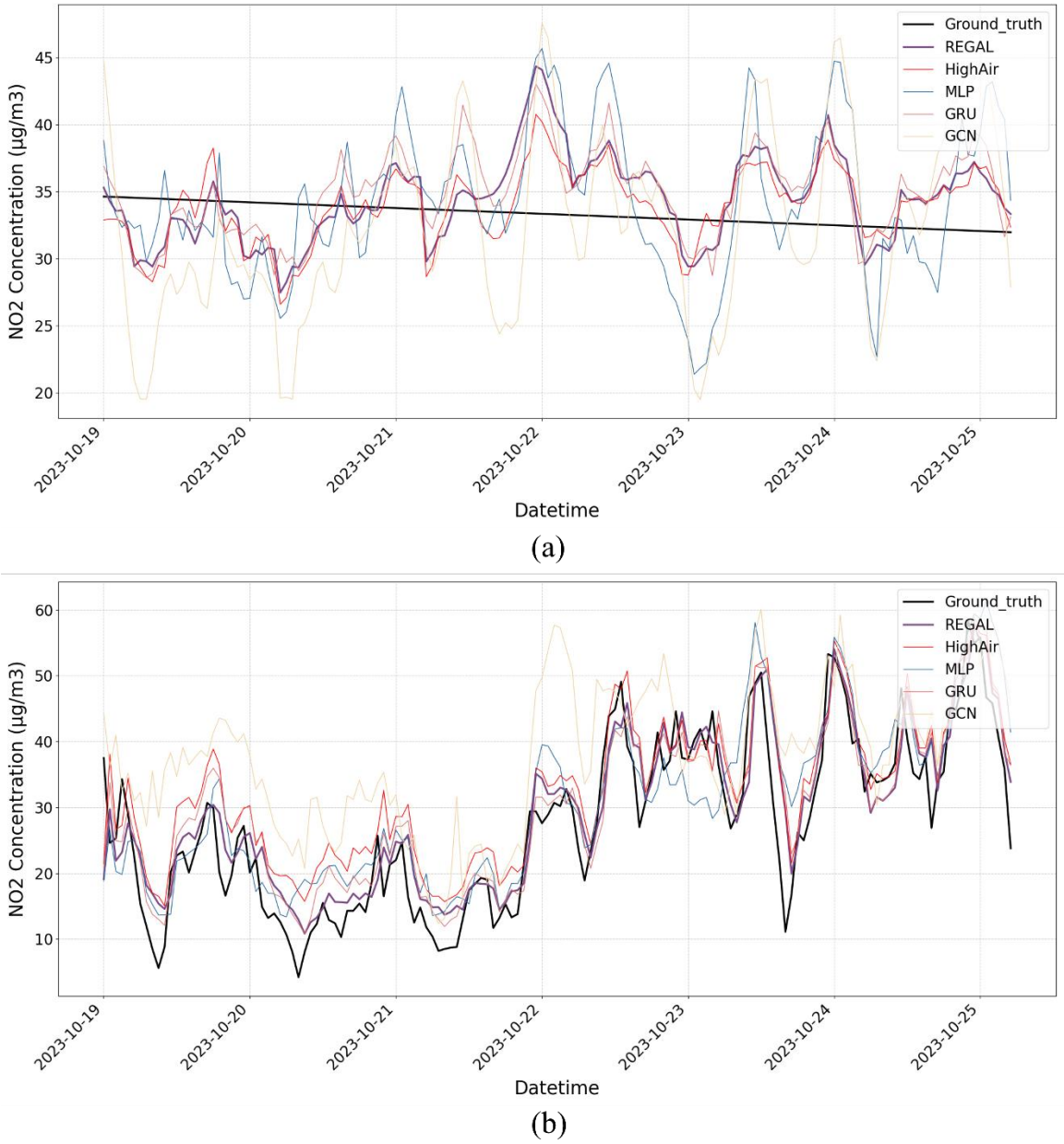

Fig. 10. Prediction fit curves and the actual time series for Station Brent-Ikea (BT4) and Lewisham (LW4)

4.3.2 Ablation Study

Ablation study is an experimental method commonly used in machine learning and deep learning to evaluate the contribution of different components or features within a model.

By systematically removing or modifying certain parts of the model and observing changes in performance, one can gain a better understanding of the significance of each component or feature (Yu et al., 2021).

First, the effectiveness of different features was examined. The conditions *w/o Road Condition*, *w/o Land Use* and *w/o Traffic Volume* represent models with the removal of Road Condition, Land Use, and Traffic Volume from the static features, respectively. Meanwhile, *w/o NO2* and *w/o Meteorology* denote the removal of historical NO2 concentration and Meteorological information from the temporal features. As shown in Figure 11 (a), removing any category of features results in increased prediction error. The removal of historical NO2 concentration has the most significant impact on prediction error, indicating a strong temporal dependency in road emissions. Traffic features also have a substantial impact, even slightly more than Meteorology data, demonstrating the effectiveness of the selected traffic features in improving model prediction. However, apart from the removal of historical NO2 concentration, removing other features does not significantly reduce the PCC metric, suggesting that the correlation in time series is primarily related to historical data.

Next, the effectiveness of model's different components was examined. *w/o Spatial* indicates the removal of the GAT module from the proposed model, meaning that no spatial information is passed among monitoring stations. *w/o Temporal* signifies the removal of the LSTM module, which means it does not learn temporal dependencies from historical data. *w/o Wind-driven* represents the removal of the module that constructs dynamic graphs based on the advection process of air pollutants. Although these three components serve different functions, to maintain a fair comparison, all types of features were input into each component for learning, with their dimensions adjusted accordingly. As shown in Figure 11 (b), each component in REGAL contributes to the model's performance, especially the *w/o Wind-driven* where the removal of only the wind-driven dynamic graph structure results in a noticeable performance decline, proving the superiority of incorporating domain knowledge into deep learning.

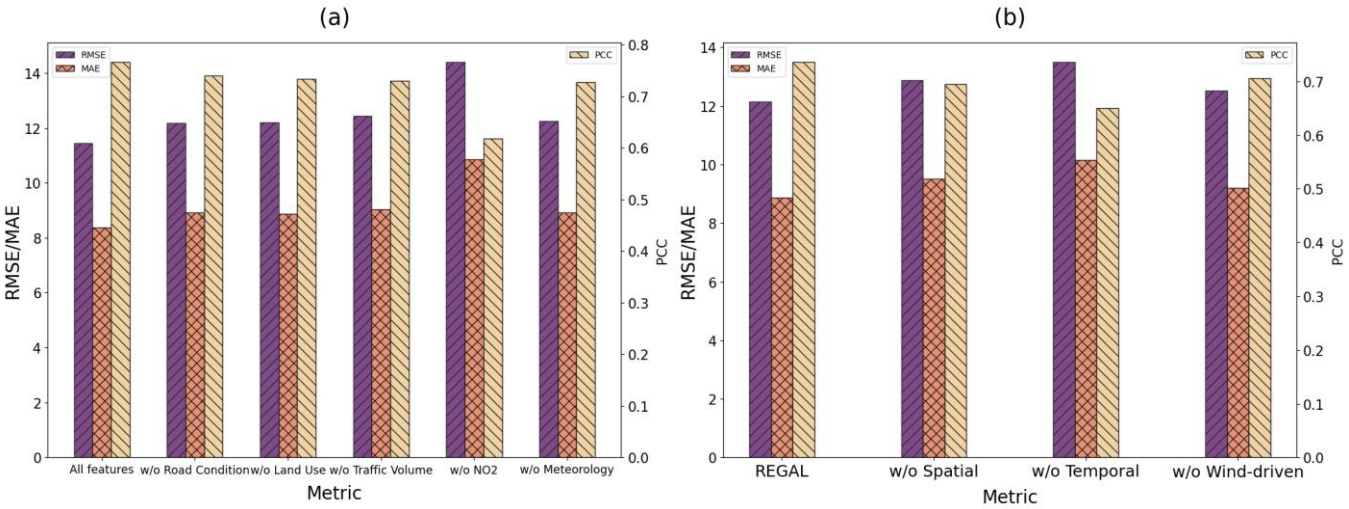

Fig. 11. Ablation experiments of the proposed model

5 Conclusion and Future Work

This paper proposes a novel spatiotemporal modeling approach based on the graph structure of urban monitoring stations to predict road NO2 emissions. First, a comprehensive collection of spatiotemporal feature data was gathered, including spatial static features such as road conditions, traffic features, and land use; temporal features like meteorological conditions (precipitation, wind speed, humidity); and global features like the calendar, which influences human emission activities. Second, meteorological knowledge was integrated to construct a spatiotemporal dynamic graph of air pollutant monitoring stations based on geographical proximity and the wind-driven advection process. The proposed REGAL (Road Emission GAT-LSTM) model combines Graph Attention Networks and LSTM to capture the spatiotemporal dependencies of various feature types effectively. To validate the model's applicability, experiments were conducted using a spatiotemporal dataset of hourly observed air pollutants from roadside monitoring stations across London. The results indicate that the REGAL model outperforms existing baseline methods in prediction accuracy. Specifically, the graph attention mechanism improves prediction accuracy by 12.5% by automatically capturing spatial correlation weights between different stations. Additionally, REGAL employs an encoder-decoder architecture that extends the forecast horizon to 12 hours. The model's advantage over other baseline models is more pronounced in long-term (9-12h) road

emission predictions than in short-term (1-4h), demonstrating its success in capturing the complex spatiotemporal dependencies underlying road emissions.

Accurately predicting spatiotemporal road emissions holds significant practical implications for a metropolis like London. The ablation study in this research highlights the considerable impact of land use, road conditions, and traffic features on road emissions, providing valuable insights for urban planning and policy design. For instance, in road planning and traffic control, optimizing road design to reduce idling emissions and encouraging the development of dedicated lanes for different vehicle types can more efficiently manage traffic flow and reduce overall emissions. In land-use planning, increasing urban green spaces near major roads can absorb NO2 and reduce pollution levels, while limiting high-density building developments can prevent the "street canyon" effect and ensure better pollutant dispersion.

In the future, we plan to extend our approach in the following areas:

1. Constructing the spatial graph structure of monitoring stations is essentially an interpretation of the spatial correlations in road emissions. Future work could incorporate more domain knowledge to construct spatiotemporal graphs of monitoring stations, such as modeling not only the advection process of air pollutants but also the diffusion process driven by pollutant concentration gradients.
2. On the data and feature extraction front, we plan to utilize APIs to retrieve real-time traffic flow and apply cosine encoding to time information (e.g., hours, months) to better capture periodic features, aiming to improve the model's prediction accuracy during peak and low emission periods.
3. We will explore the use of semi-supervised learning to extend road emission predictions to areas without monitoring stations or where data is sparse.
4. Additionally, we will adopt interpretable deep learning frameworks to uncover the specific contributions and spatiotemporal heterogeneity of factors influencing road emissions.

Reference

[1] Beevers, S. D., Westmoreland, E., de Jong, M. C., Williams, M. L., & Carslaw, D. C. (2012). Trends in NOx and NO2 emissions from road traffic in Great Britain. Atmospheric Environment, 54, 107–116. https://doi.org/10.1016/j.atmosenv.2012.02.028

[2] Boeing, G. (2017). OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. Computers, Environment and Urban Systems, 65, 126–139. https://doi.org/10.1016/j.compenvurbsys.2017.05.004

[3] Chelani, A. B., & Devotta, S. (2006). Air quality forecasting using a hybrid autoregressive and nonlinear model. Atmospheric Environment, 40(10), 1774–1780. https://doi.org/10.1016/j.atmosenv.2005.11.019

[4] Chen, F.-W., & Liu, C.-W. (2012). Estimation of the spatial rainfall distribution using inverse distance weighting (IDW) in the middle of Taiwan. Paddy and Water Environment, 10(3), 209–222. https://doi.org/10.1007/s10333-012-0319-1

[5] Chen, L., Ding, Y., Lyu, D., Liu, X., & Long, H. (2019). Deep Multi-Task Learning Based Urban Air Quality Index Modelling. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 3(1), 2:1-2:17. https://doi.org/10.1145/3314389

[6] Dey, R., & Salem, F. M. (2017). Gate-variants of Gated Recurrent Unit (GRU) neural networks. 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), 1597–1600. https://doi.org/10.1109/MWSCAS.2017.8053243

[7] Ge, L., Wu, K., Zeng, Y., Chang, F., Wang, Y., & Li, S. (2021). Multi-scale spatiotemporal graph convolution network for air quality prediction. Applied Intelligence, 51(6), 3491–3505. https://doi.org/10.1007/s10489-020-02054-y

[8] Han, J., Liu, H., Xiong, H., & Yang, J. (2023). Semi-Supervised Air Quality Forecasting via Self-Supervised Hierarchical Graph Neural Network. IEEE Transactions on Knowledge and Data Engineering, 35(5), 5230–5243. IEEE Transactions on Knowledge and Data Engineering. https://doi.org/10.1109/TKDE.2022.3149815

[9] Hettige, K. H., Ji, J., Xiang, S., Long, C., Cong, G., & Wang, J. (2024). AIRPHYNET: HARNESSING PHYSICS-GUIDED NEU- RAL NETWORKS FOR AIR QUALITY PREDICTION.

[10] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. Neural Computation. https://doi.org/10.1162/neco.1997.9.8.1735

[11] Hood, C., MacKenzie, I., Stocker, J., Johnson, K., Carruthers, D., Vieno, M., & Doherty, R. (2018). Air quality simulations for London using a coupled regional-to-local modelling system. Atmospheric Chemistry and Physics, 18(15), 11221–11245. https://doi.org/10.5194/acp-18-11221-2018

[12] Huang, C.-J., & Kuo, P.-H. (2018). A Deep CNN-LSTM Model for Particulate Matter (PM2.5) Forecasting in Smart Cities. Sensors (Basel, Switzerland), 18(7), 2220. https://doi.org/10.3390/s18072220

[13] Iskandaryan, D., Ramos, F., & Trilles, S. (2023). Graph Neural Network for Air Quality Prediction: A Case Study in Madrid. IEEE Access, 11, 2729–2742. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3234214

[14] Jin, X.-B., Wang, Z.-Y., Kong, J.-L., Bai, Y.-T., Su, T.-L., Ma, H.-J., & Chakrabarti, P. (2023). Deep Spatio-Temporal Graph Network with Self-Optimization for Air Quality Prediction. Entropy, 25(2), 247. https://doi.org/10.3390/e25020247

[15] Kim, D.-Y., Jin, D.-Y., & Suk, H.-I. (2023). Spatiotemporal graph neural networks for predicting mid-to-long-term PM2.5 concentrations. Journal of Cleaner Production, 425, 138880. https://doi.org/10.1016/j.jclepro.2023.138880

[16] Martin, R. V. (2008). Satellite remote sensing of surface air quality. Atmospheric Environment, 42(34), 7823–7843. https://doi.org/10.1016/j.atmosenv.2008.07.018

[17] Oliveira Santos, V., Costa Rocha, P. A., Scott, J., Van Griensven Thé, J., & Gharabaghi, B. (2023). SHAP·Spatiotemporal Air Pollution Forecasting in Houston-TX: A Case Study for Ozone Using Deep Graph Neural Networks. Atmosphere, 14(2), Article 2. https://doi.org/10.3390/atmos14020308

[18] Ouyang, X., Yang, Y., Zhang, Y., & Zhou, W. (2021). Spatial-Temporal Dynamic Graph Convolution Neural Network for Air Quality Prediction. 2021 International Joint Conference on Neural Networks (IJCNN), 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534167

[19] Prieto-Rodriguez, J., Perez-Villadoniga, M. J., Salas, R., & Russo, A. (2022). Impact of London Toxicity Charge and Ultra Low Emission Zone on NO2. Transport Policy, 129, 237–247. https://doi.org/10.1016/j.tranpol.2022.10.010

[20] Shiva Nagendra, S. M., & Khare, M. (2004). Artificial neural network based line source models for vehicular exhaust emission predictions of an urban roadway. Transportation Research Part D: Transport and Environment, 9(3), 199–208. https://doi.org/10.1016/j.trd.2004.01.002

[21] Smit, R., Ntziachristos, L., & Boulter, P. (2010). Validation of road vehicle and traffic emission models – A review and meta-analysis. Atmospheric Environment, 44(25), 2943–2953. https://doi.org/10.1016/j.atmosenv.2010.05.022

[22] van Donkelaar, A., Martin, R. V., & Park, R. J. (2006). Estimating ground-level PM2.5 using aerosol optical depth determined from satellite remote sensing. Journal of Geophysical Research: Atmospheres, 111(D21). https://doi.org/10.1029/2005JD006996

[23] Vardoulakis, S., Fisher, B. E. A., Pericleous, K., & Gonzalez-Flesca, N. (2003). Modelling air quality in street canyons: A review. Atmospheric Environment, 37(2), 155–182. https://doi.org/10.1016/S1352-2310(02)00857-9

[24] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2017, October 30). Graph Attention Networks. arXiv.Org. https://arxiv.org/abs/1710.10903v3

[25] Wang, C., Zhu, Y., Zang, T., Liu, H., & Yu, J. (2021). Modeling Inter-station Relationships with Attentive Temporal Graph Convolutional Network for Air Quality Prediction. Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 616–634. https://doi.org/10.1145/3437963.3441731

[26] Wang, S., Qiao, L., Fang, W., Jing, G., S. Sheng, V., & Zhang, Y. (2022). Air Pollution Prediction Via Graph Attention Network and Gated Recurrent Unit. Computers, Materials & Continua, 73(1), 673–687. https://doi.org/10.32604/cmc.2022.028411

[27] Xu, J., Chen, L., Lv, M., Zhan, C., Chen, S., & Chang, J. (n.d.). HighAir: A Hierarchical Graph Neural Network-Based Air Quality Forecasting Method.

[28] Xu, J., Wang, S., Ying, N., Xiao, X., Zhang, J., Jin, Z., Cheng, Y., & Zhang, G. (2023). Dynamic graph neural network with adaptive edge attributes for air quality prediction: A case study in China. Heliyon, 9(7), e17746. https://doi.org/10.1016/j.heliyon.2023.e17746

[29] Xu, Z., Cao, Y., & Kang, Y. (2019). Deep spatiotemporal residual early-late fusion network for city region vehicle emission pollution prediction. Neurocomputing, 355, 183–199. https://doi.org/10.1016/j.neucom.2019.04.040

[30] Yu, L., Du, B., Hu, X., Sun, L., Han, L., & Lv, W. (2021). ☆ Deep spatio-temporal graph convolutional network for traffic accident prediction. Neurocomputing, 423, 135 – 147. https://doi.org/10.1016/j.neucom.2020.09.043

[31] Zeng, J., Liu, Y., Ding, J., Yuan, J., & Li, Y. (2024). Estimating On-Road Transportation Carbon Emissions from Open Data of Road Network and Origin-Destination Flow Data. Proceedings of the AAAI Conference on Artificial Intelligence, 38(20), 22493–22501. https://doi.org/10.1609/aaai.v38i20.30257

[32] Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks: A comprehensive review. Computational Social Networks, 6(1), 11. https://doi.org/10.1186/s40649-019-0069-y

[33] Zhang, X., & Gong, Z. (2018). Spatiotemporal characteristics of urban air quality in China and geographic detection of their determinants. Journal of Geographical Sciences, 28(5), 563–578. https://doi.org/10.1007/s11442-018-1491-z

[34] Zhao, P., & Zettsu, K. (2020). MASTGN: Multi-Attention Spatio-Temporal Graph Networks for Air Pollution Prediction. 2020 IEEE International Conference on Big Data (Big Data), 1442–1448. https://doi.org/10.1109/BigData50022.2020.9378156

[35] Zheng, Y., Liu, F., & Hsieh, H.-P. (2013). U-Air: When urban air quality inference meets big data. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1436–1444. KDD' 13: The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/2487575.2488188Zhao, P., & Zettsu, K. (2020). MASTGN: Multi-Attention Spatio-Temporal Graph Networks for Air Pollution Prediction. 2020 IEEE International Conference on Big Data (Big Data), 1442–1448. https://doi.org/10.1109/BigData50022.2020.9378156

[36] Zheng, Y., Liu, F., & Hsieh, H.-P. (2013). U-Air: When urban air quality inference meets big data. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1436–1444. KDD' 13: The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/2487575.2488188