

Article

# Embedding-Graph-Neural-Network for Transient NO<sub>x</sub> Emissions Prediction

Yun Chen <sup>1</sup>, Chengwei Liang <sup>1</sup>, Dengcheng Liu <sup>2</sup>, Qingren Niu <sup>1</sup>, Xinke Miao <sup>1</sup> , Guangyu Dong <sup>1,\*</sup>, Liguang Li <sup>1</sup>, Shanbin Liao <sup>3</sup>, Xiaoci Ni <sup>1</sup> and Xiaobo Huang <sup>3</sup>

<sup>1</sup> School of Automotive Studies, Tongji University, Shanghai 201804, China

<sup>2</sup> Nanchang Automotive Institute of Intelligence & New Energy, Nanchang 330001, China

<sup>3</sup> Jiangling Motors Corporation, Nanchang 330001, China

\* Correspondence: g.dong@tongji.edu.cn

**Abstract:** Recently, Acritical Intelligent (AI) methodologies such as Long and Short-term Memory (LSTM) have been widely considered promising tools for engine performance calibration, especially for engine emission performance prediction and optimization, and Transformer is also gradually applied to sequence prediction. To carry out high-precision engine control and calibration, predicting long time step emission sequences is required. However, LSTM has the problem of gradient disappearance on too long input and output sequences, and Transformer cannot reflect the dynamic features of historic emission information which derives from cycle-by-cycle engine combustion events, which leads to low accuracy and weak algorithm adaptability due to the inherent limitations of the encoder-decoder structure. In this paper, considering the highly nonlinear relation between the multi-dimensional engine operating parameters the engine emission data outputs, an Embedding-Graph-Neural-Network (EGNN) model was developed combined with self-attention mechanism for the adaptive graph generation part of the GNN to capture the relationship between the sequences, improve the ability of predicting long time step sequences, and reduce the number of parameters to simplify network structure. Then, a sensor embedding method was adopted to make the model adapt to the data characteristics of different sensors, so as to reduce the impact of experimental hardware on prediction accuracy. The experimental results show that under the condition of long-time step forecasting, the prediction error of our model decreased by 31.04% on average compared with five other baseline models, which demonstrates the EGNN model can potentially be used in future engine calibration procedures.

**Keywords:** LSTM; transformer; sparse graph attention; EGNN



**Citation:** Chen, Y.; Liang, C.; Liu, D.; Niu, Q.; Miao, X.; Dong, G.; Li, L.; Liao, S.; Ni, X.; Huang, X. Embedding-Graph-Neural-Network for Transient NO<sub>x</sub> Emissions Prediction. *Energies* **2023**, *16*, 3. <https://doi.org/10.3390/en16010003>

Received: 5 November 2022

Revised: 13 December 2022

Accepted: 16 December 2022

Published: 20 December 2022



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the last decades, the greenhouse effect has gradually become a global issue [1], and higher thermal efficiency and lower emission output have become the primary goals for the development of modern internal combustion engines. Correspondingly, emission regulations are becoming increasingly stringent to reduce emissions in the transportation section of each country [2,3]. Automotive manufacturers need to meet these regulations under complex transient test conditions, and to minimize vehicle emissions. In fact, there is a significant difference between the steady-state test and transient test of engine emission performance [4]. Steady-state data usually means the average output value of the engine over a period. Due to the low time resolution of conventional emission analyzers, the results of steady-state experiments normally cannot compare with real-time engine performance. Fast response emissions analyzers can collect transient data to measure emissions variations under transient operating conditions such as engine start/stop and acceleration/deceleration. Transient operating conditions normally represent the real engine working conditions, sometimes with an extremely high-resolution ratio. Therefore,

as an assessment tool, an accurate engine emission prediction model, which can be applied to evaluate the engine transient emission performance, is critical for engine control and calibration.

In view of the deficiency of online NO<sub>x</sub> monitoring equipment in practical application, CFD (Computational Fluid Dynamics) simulations [5–7] are commonly used to predict NO<sub>x</sub> emissions and have been applied to optimize the engine control strategies in the past. However, for real-time forecasting demand, CFD is not applicable due to its high computational cost, high calibration requirements, complex structure [8–11] and poor adaptability to different engines. Considering the high dimensional engine control MAPs and lookup tables which are embedded in the engine ECU system, the number of experiments required is exponential, which leads to excessive cost. Recently, Acritical Intelligent (AI) methodologies have been widely considered as promising tools for engine performance calibration, especially for engine emission performance prediction and optimization. Liu et al. [12] combined a data preprocessing and long and short-term memory (LSTM) model [13] to estimate numerical emission predictions under unsteady conditions with acceptable accuracy. Kiyas et al. [14] modeled combustion efficiency and exhaust emission indexes for a turbocharged engine with an LSTM model. Halil et al. [15] compared a Deep Neural Network (DNN) to a traditional artificial neural network (ANN) [16–18] method regarding the emission prediction ability. The results showed that the performance of the DNN model was much better than the traditional ANN model with a lower relative error. However, the above models are only applicable to single time-step emission prediction, and their multi-time-step forecasting performances are limited.

To achieve a high-precision engine control and calibration method, predicting long time step emission sequences is required [19]. Emission prediction can be viewed as a time series forecasting problem and a long sequence time-series forecasting problems. In the last decade, performance degradation has proved to be the major limitation of the LSTM model due to gradient disappearance [20]. On the other hand, the Transformer model [21] has made breakthroughs in sequence events prediction [22,23]. However, both models ignore the interconnection between multivariate time series and cannot capture the information of high frequency variation characteristics. In addition, the quadratic computation process for solving the self-attention mechanism leads to a further limitation of the transformer model when predicting a long sequence [24]. The potential solution for handling the above issue is a graph neural network (GNN) model [25], since it enhances the spatial modeling and spatiotemporal feature extraction ability. GNN is based on the graph convolution structure, which has the possibility to avoid the gradient disappearance problem.

Based on the above analysis, the main purpose of this study was to create a general method for engine NO<sub>x</sub> emission prediction with high prediction accuracy. We propose a novel graph neural network model based on dimensional embedding algorithm (EGNN), and the performance of this model is and conventional GNN model. The results show that the present EGNN model yields higher accuracy and better performance. A fast emissions analyzer is utilized in this study for data acquisition. Compared with the steady-state emission meter, fast emissions analyzers can obtain more data in a given time period and save the experimental cost.

To achieve such a goal, the main tasks of this work can be described as follows:

- (1) First, attaching a multi-dimensional variable encoding layer to the attention map learning layer. Since there are signal characteristic differences between different sensors such as data accuracy, digital resolution and signal drifting level, an encoder layer is applied to deal with these differences and embed them into the network as a new data feature. In addition, the application of the encoding layer guarantees the EGNN model a better anti-noise performance comparing to other models.
- (2) Second, a sparse attention method is coupled with the self-attention graph generation mechanism to convert the high-dimensional graph relationships into low-dimensional ones, thereby the reduction of model inputs and memory overhead becomes possible [26]. In addition, the proposed attention mechanism provides the ability to

accurately predict the long-time step sequence. Compared with the traditional GNN, the self-attention mechanism generates a graph structure automatically, which saves the process of manually finding the relationship between variables.

- (3) To improve the data accuracy, transient emitters are applied in the EGNN model rather than those steady emitters. Hence, the time cost can be significantly reduced when large dataset being processed.

The article is organized as follows. The related work is presented in Section 2. The structure of the proposed method is introduced in Section 3. In Section 4, the basic parameters of the engine, the bench test process and the method of data preprocessing, are described. We compare the proposed method with five other models in Section 5, and the paper is concluded in Section 6.

## 2. Related Works

In this section, studies related to engine emission prediction based on machine learning methods in the past 3 years are summarized for better understanding the background of the present study.

Mohammad et al. [27] utilized three machine learning methods to predict engine emissions, namely LSTM, ANN and random forest, and better results were achieved compared to traditional emission predicting methods. Aran et al. [28] used principal component analysis (PCA) in an emission forecasting model to reduce the number of input variables. The results showed that the predicting performance of the model can be improved in a condition in which the computational budget is limited. On the other hand, Armin et al. [29] studied the effect of different input variables on an SVM model under four different engine steady operating conditions. It was found that a model with multiple input variables yielded higher accuracy. Ma et al. [30] combined ANN and a particle swarm optimization algorithm, and the engine fuel consumption and emission performance were well predicted. The above research mainly focused on steady-state conditions, and the methods used were mainly traditional ANN and machine learning models. Our paper considers both the engine steady-state and transient-state conditions.

Fang et al. [31] applied a fast response emission analyzer in their study. The results showed that a transient model could be developed and the prediction of transient NO<sub>x</sub> emission became possible. To reduce the dimension of input variables in the emission predicating model, Nick et al. [32] tested two filtering approaches, a *p*-value test method, and a Pearson correlation coefficient method. Yu et al. [33] combined LSTM and filtering methods to predict NO<sub>x</sub> emissions under engine transient conditions. Compared with the studies listed in the literature [31–33], the major novelty of our paper is that for the first time, the possibility of modeling long-term emission sequence with fewer variables is demonstrated and verified, and the accuracy of the model developed in this work is significantly improved.

## 3. Research Methods

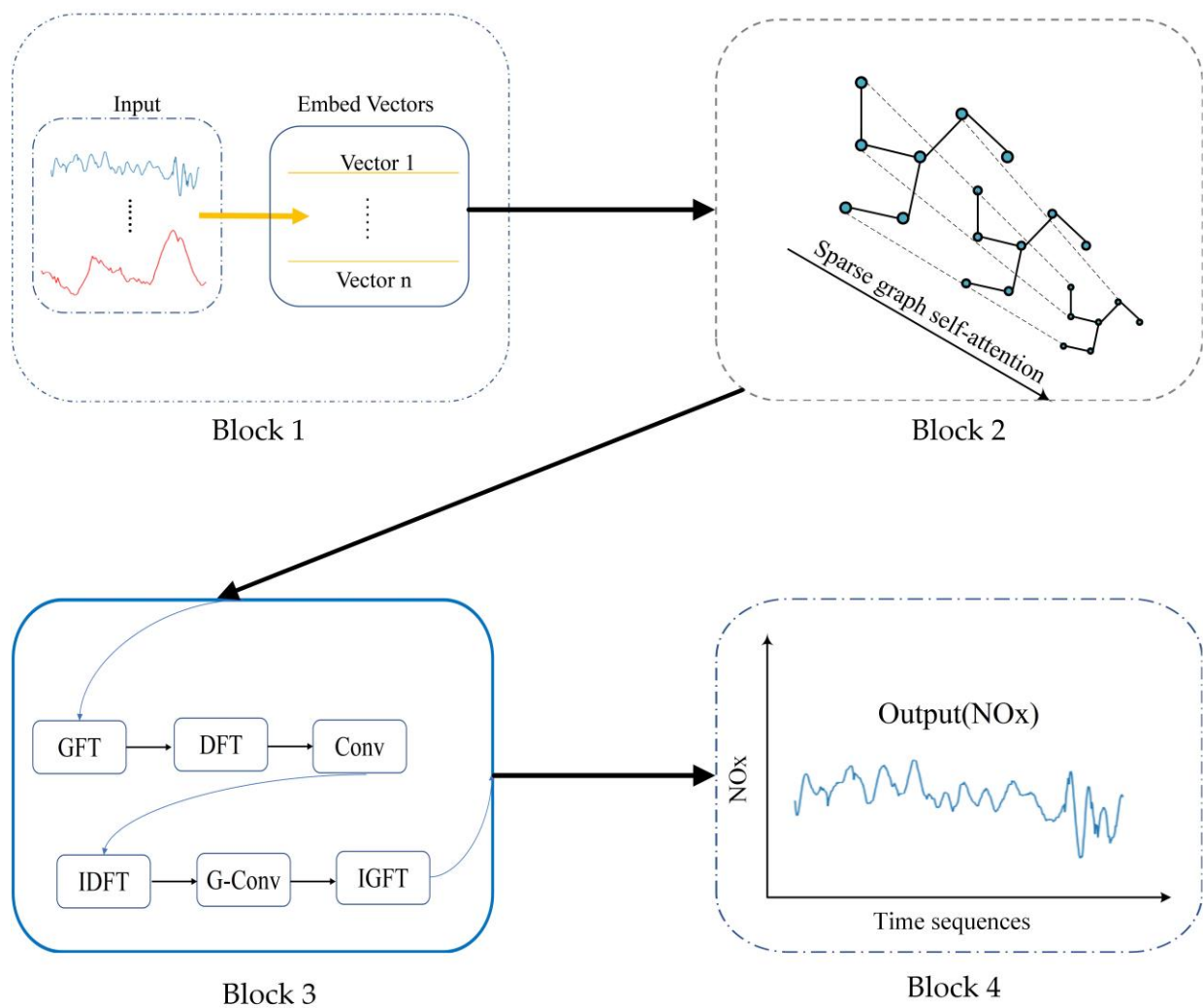
Figure 1 is a schematic diagram of our proposed structure, and involves four main blocks introduced in the following sections.

Multidimensional Variable Embedding (Section 3.2): process the variables by the embedding method to learn the unique characteristics of sensors.

Sparse graph self-attention mechanism (Section 3.3): develop a graph structure adaptively based on the improved self-graph-attention mechanism for the neural network.

Graph neural network based on Fourier transform (Section 3.4): adopt spectrogram GNN to capture the relationship in the graph structure and complete the establishment of the emission prediction model.

Output block (Section 3.4): output the prediction result of NO<sub>x</sub>.



**Figure 1.** Schematic diagram of EGNN consisting of four blocks: multidimensional variable embedding, sparse graph self-attention mechanism, graph neural network based on Fourier transform and output block.

### 3.1. Problem Definition

The NO<sub>x</sub> emission prediction is based on the time series of historical emission information to estimate the emission values in the future, which means the emission values of time  $t + \Delta t$  can be predicted at time  $t$ , where  $\Delta t$  denotes time interval between two points. To capture the relationship between multivariate variables and NO<sub>x</sub>, the data structure, multivariate graph shown in Figure 2 is adopted.



**Figure 2.** Multivariate graph structure in which each node can be connected to every other.

We define the multivariate variables model as a graph  $G = (X, A)$ , where  $X = \{x_1, x_2, \dots, x_N\}$  is a finite set of  $N$  nodes that represents multivariate variables and  $x_i \in \mathbb{R}^{m \times T}$  where  $T$  is the timestamp and  $m$  is the embedding dimension, which will be explained in the Section 3.3.  $A^{N \times N}$  represents the adjacency matrix of graph  $G$ .

With the input values of previous  $L$  time steps  $X_{t-L}, X_{t-L+1}, \dots, X_t$ , our target is to predict the  $K$  time step values of the NOx emission. The prediction can be denoted by  $X_{t+1}, X_{t+2}, \dots, X_{t+h}$ , and inferred by the deep learning model  $Y(G, X_{t-L}, X_{t-L+1}, \dots, X_t)$ .

### 3.2. Multidimensional Variable Embedding

Multidimensional variables represent multiple characteristic inputs to the model. Various sensors have their own unique characteristics in different stages and environments. At the same time, their initial noise values are different. To reduce the influence of hardware on the model’s prediction performance, we utilized an encoding method to generate feature vectors for multidimensional variables. Compared with the positional encoding in Transformer [21], the embedding vectors provided to the input variables are learnable.

Figure 3 is a schematic diagram of encoding. The representation method for embedding vector is as follows:

$$V_i = (\omega_{i1}, \omega_{i2}, \dots, \omega_{id}) \text{ for } i \in \{1, 2, \dots, N\}, d \in N^* \tag{1}$$

where  $N$  is the number of the variables and  $N^*$  is positive integer set.

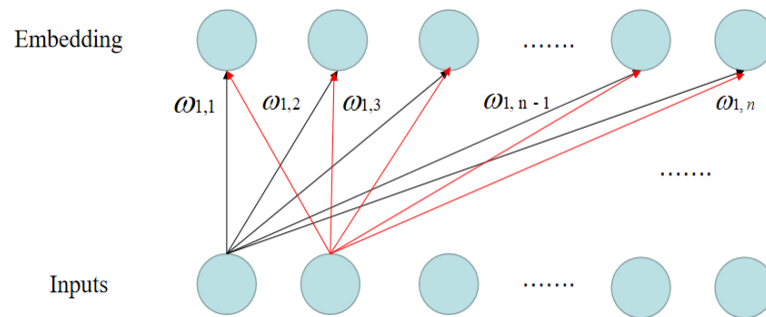


Figure 3. Embedding process for multidimensional variables.

### 3.3. Sparse Graph Self-Attention Mechanism

#### 3.3.1. Sparse Self-Attention Mechanism

In the Transformer [26], the self-attention calculates the dot product of the  $i$ -th query as follows:

$$ST(q_i, K, V) = (q_i K^T / \sqrt{d}) V \tag{2}$$

where  $d$  is the model’s input dimension,  $q_i$  is the  $i$ -th row of  $Q$ , and  $Q, K, V \in \mathbb{R}^{L \times d}$ .  $L$  is the input sequence length.

In the self-attention mechanism, according to the Formula (2), the probability distribution of  $Q$  and  $K$  is as follows:

$$p(k_j | q_i) = K(q_i, k_j) / \sum_l K(q_i, k_l) \tag{3}$$

$K(q_i, k_j)$  is the asymmetric exponential kernel  $\exp(q_i k_j^T / \sqrt{d})$ ,  $q_i, k_j$  representing column  $i$  of  $Q$  matrix and column  $j$  of  $K$  matrix, respectively.

We define a uniform distribution as  $x(k_j | q_i) = 1/L$ . When the matrix probability distribution of  $Q$  and  $K$  approximates a uniform distribution, then, for the overall self-attention mechanism, the weight is approximated to a random input, which means the neural network loses its function. Therefore, we adopted the MMD method (Maximum

Mean Discrepancy) shown in Formula (5) to measure the similarity between parameter weight distribution and uniform distribution.

$$d_k^2(p, q) = \| E_{x \sim p}[\Phi(x)] - E_{x \sim q}[\Phi(x)] \|_{HK}^2 \tag{4}$$

For the convenience of calculation, Formula (4) is transformed into the following form, and the linear kernel function is used for mapping:

$$U(p, q) = \frac{1}{L^2} (e^{\frac{q_i k_j^T}{\sqrt{d}}} \sum_{a=1}^L e^{\frac{q_i k_a^T}{\sqrt{d}}} + \sum_{a=1}^L (\frac{1}{L^2} - 2e^{\frac{q_i k_a^T}{\sqrt{d}}})) \tag{5}$$

Multiply both sides of the above equation by  $L^2$  and drop the constant to get the Formula (6):

$$U(p, q) = (e^{\frac{q_i k_j^T}{\sqrt{d}}} - 2) \sum_{a=1}^L e^{\frac{q_i k_a^T}{\sqrt{d}}} \tag{6}$$

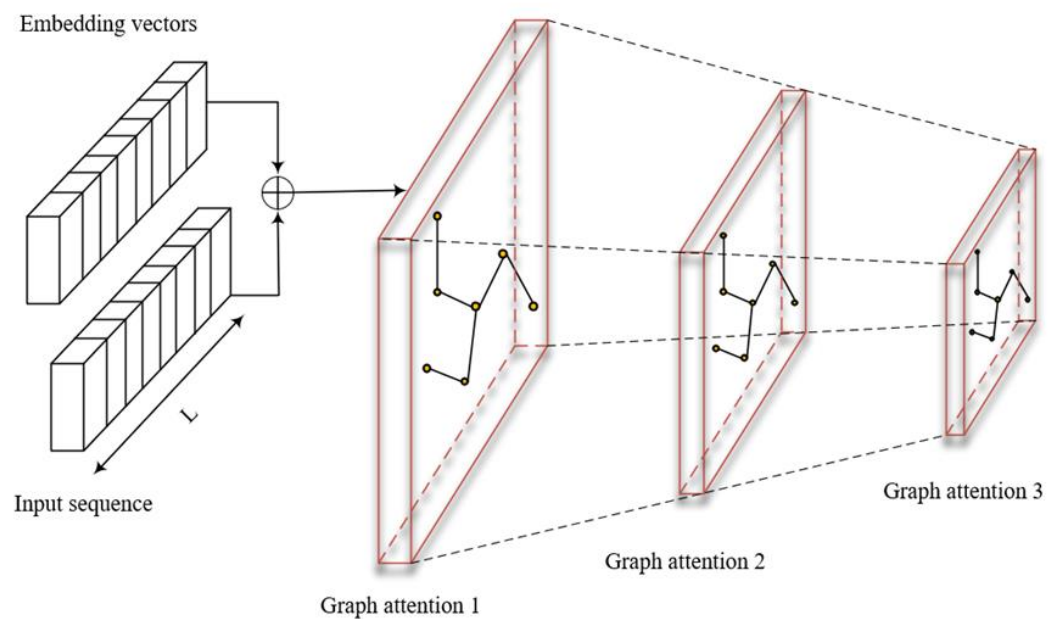
Following the inference in the informer [28], which is  $\ln \sum_{a=1}^L e^{\frac{q_i k_a^T}{\sqrt{d}}} \approx \max(\frac{q_i k_a^T}{\sqrt{d}})$ , the Formula (6) can be rewritten as the following form:

$$U(p, q) = e^{\frac{q_i k_j^T}{\sqrt{d}}} \cdot \max_j (e^{\frac{q_i k_a^T}{\sqrt{d}}}) - 2 \sum_{a=1}^n e^{\frac{q_i k_a^T}{\sqrt{d}}} \tag{7}$$

By Formula (7), dot product pairs with higher attention weights are screened out to make the size of learned graph smaller, with the depth of the model reduced concerning parameters memory consumption as shown in Figure 4. Therefore, the matrix  $Q$  is transformed into the sparse special matrix in Formula (8). The hyperparameter  $s$  is set that indicates the Top- $s$  product pairs are selected as the input of the next layer.

$$Q^* = sam(Q|u(p, q)) \tag{8}$$

where  $sam(\cdot)$  is the function that samples the  $q$  in top- $s$  product pairs to generate the  $Q^* \in \mathbb{R}^{s \times d}$ .



**Figure 4.** Sparse graph self-attention mechanism. Embedding vector represents multidimensional variable embedding.

### 3.3.2. Graph Construction

Simple GNN-based methods require a graph structure when modeling engine multi-dimensional parameters. The graph structure can be constructed by physical knowledge or experience, which can be difficult to build because of the nonlinear relationship among the parameters of the engine. To adapt to different engines, we utilize the sparse self-attention mechanism to automatically learn the potential correlations between multiple engine parameters as shown in Formula (9):

$$G = \text{Softmax}(\text{relu}(KQ^* / \sqrt{d})V) \quad (9)$$

where  $Q^*$  is the sparse spatial matrix generated in Formula (8),  $\text{relu}$  function is chosen as the activation function, and  $\text{Softmax}$  is adopted to normalize the generated graph.

In this way, the model builds a graph structure of its own parameters in a data-driven manner.

### 3.4. Graph Neural Network Based on Fourier Transform

Most time series research has focused on the time-domain correlation between multi-dimensional variables, while ignoring the intrinsic relationship between the parameters in the frequency domain. Considering the high-frequency variation of parameters caused by high engine speed, we attempted to use an improved spectrogram neural network to establish an engine emission prediction model by combining the graph generated in the Block 2 of Figure 1. The improved spectrogram neural network Block is displayed in the block 3 of Figure 1.

To combine the sensor embedding vectors  $v_i$  with the neural network, we improved the overall calculation flow. First, our graph generator incorporates  $v_i$  and to do this, we compute the graph attention as follows:

$$z_i^{(t)} = v_i + x_i^{(t)} \quad (10)$$

$$h(i, j) = \text{Gelu}(w_q z_i \cdot w_k z_j) \quad (11)$$

$$\alpha_{i,j} = \frac{\exp(h(i, j))}{\sum \exp(h(i, k))} \quad (12)$$

where Formula (10) combines the input features and embedding vectors, and  $\text{Gelu}$  is used as the nonlinear activation to calculate the attention coefficient in Formula (11).  $\text{Softmax}$  function normalizes the attention coefficients in Formula (12).

The generated emission graph is transformed into a spectral matrix by Graph Fourier Transform (GFT), which maps the input graph to an orthonormal space based on the normalized graph Laplacian. The normalized graph Laplacian  $Lap$  is shown in Formula (13):

$$Lap = I - D^{-0.5}GD^{-0.5} \quad (13)$$

where  $I$  is the identity matrix,  $D$  is a degree matrix and  $G$  is the graph generated in Formula (9).

In Formula (14), the  $Lap$  can be spectrally decomposed, the matrix composed of its eigenvectors is  $U$ , and  $A$  is the eigenvalue matrix.

$$Lap = UAU^T \quad (14)$$

Therefore, GFT and Inverse Graph Fourier Transform (IGFT) can be calculated as shown in Formulas (15) and (16).

$$F(X) = U^T X = \hat{X} \quad (15)$$

$$F^{-1}(\hat{X}) = U^T \hat{X} \quad (16)$$

After the operation, the multi-dimensional sensor variables become independent of each other. Then, by the Discrete Fourier transform (DFT) operation, variables for the engine are transformed into the frequency domain, which can capture feature information under the Gated Linear Units (GLU) layers' transform and the convolution layer. Inverse Discrete Fourier Transform (IDFT) brings the variables back to the time domain. Finally, we use the graph convolution in Formula (17) and IGFT to generate the final output.

$$G - conv = \sum_{k=0}^K \theta_k (UAU^T)^k \quad (17)$$

where  $\theta_k$  is the weight of the k-hop neighbor.

#### 4. Engine Bench and Data Preparation

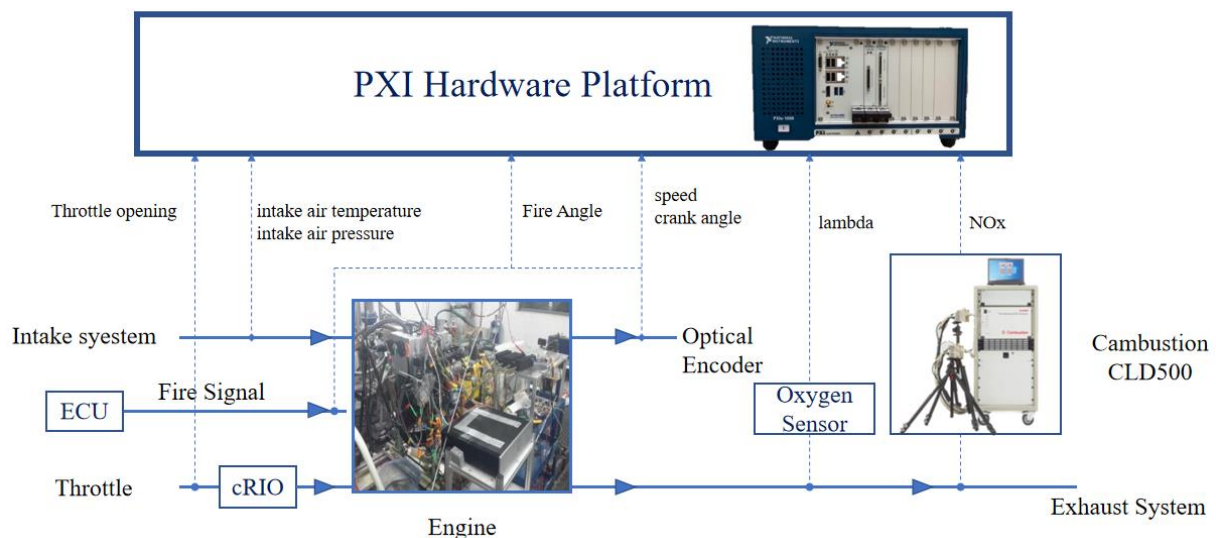
To measure the prediction performance of EGNN, we conducted multiple sets of experiments on an engine with model JLB-4G14TB, and obtained a dataset called EDT containing 140,000 sample points.

##### 4.1. Test Bench

The research was based on a four-cylinder, four-stroke, gasoline engine, whose specifications are presented in Table 1. A schematic diagram of the test bench is shown in Figure 5.

**Table 1.** Engine specifications Engine specifications.

Engine Parameters	Values
Intake mode	Turbocharger
Number of cylinders	4
Maximum power rotational speed	5200 rpm
Maximum torque rotational speed	1600–4000 rpm
Displacement	1.4 L
Valves per cylinder	4
Valve system	DOHC
Maximum power	104 kW
Maximum torque	235 N.m
Cylinder head material	Aluminum alloy



**Figure 5.** Schematic diagram of the test bench. The bench is composed of the engine, the dynamometer, data acquisition system PXI, CLD500 fast analyzer and other sensors.

In the test bench, the intake pressure and exhaust-gas pressure were measured by a CYYZ11A pressure transmitter. The intake temperature and the exhaust-gas temperature were measured by PT100 and a thermocouple temperature sensor, respectively. The torque was controlled by a motor which connected the CompactRIO control system. The NOx emission was measured by a Cambustion CLD 500 transient emission analyzer. All sensors were connected to the PXI hardware platform for data acquisition. The model and measurement accuracy of the experimental equipment are shown in Table 2.

**Table 2.** Measuring devices of the engine test bench.

Equipment Name	Type	Precision
Thermocouple temperature sensor	WRN-230	±0.5 °C
Air intake temperature sensor	PT100	±0.15 °C
Photoelectric encoder	A-CHA	±0.1°
Transient emission analyzer	CLD 500	NOx:1 ppm
Pressure Transmitter	CYYZ11A	±0.1% FS
Oxygen Sensor	LSU4.9	±1% FS

#### 4.2. Work Conditions

In the experiments, engine load was changed from 6 to 100% full load in 2 to 20% intervals by controlling the dynamometer. At the beginning, the speed was controlled at 800 r/min for ten minutes to warm up the lubricating oil. After that, the engine started to gradually accelerate from idle speed to 3500 r/min. During the period, rapid deceleration and rapid acceleration were randomly performed to simulate braking and random acceleration in real road driving conditions. To ensure the amount of data under steady-state conditions, we sampled at intervals of 250 r/min in the speed range from 1000 rpm to 3000 rpm. The scanning time of each interval was not more than 15 s. In addition, we scanned different speeds at the same load condition. The load variation range was 6 to 50%, while the speed range was 1000 rpm to 3000 rpm. The data were collected using the crankshaft angle as a trigger and data collecting was triggered every 180° turn.

#### 4.3. Data Preparation

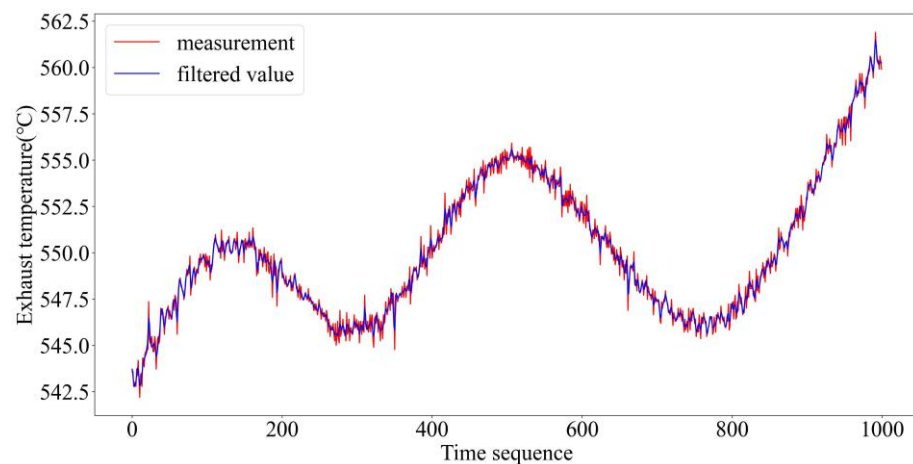
The data were filtered using the Complete Ensemble Empirical Mode Decomposition with Adaptive Noise method (CEEMDAN) [34]. CEEMDAN makes up for the shortcomings of the EMD method and often be used for high-frequency filtering processing of nonlinear signals. When generating each IMF (Intrinsic Mode Function), white noise is added to reduce the modal aliasing of the EMD decomposition, thereby reducing the reconstruction error.

The partial results of filtering for variables are shown in Figures 6 and 7. The intake air pressure was calculated by using the relative pressure.

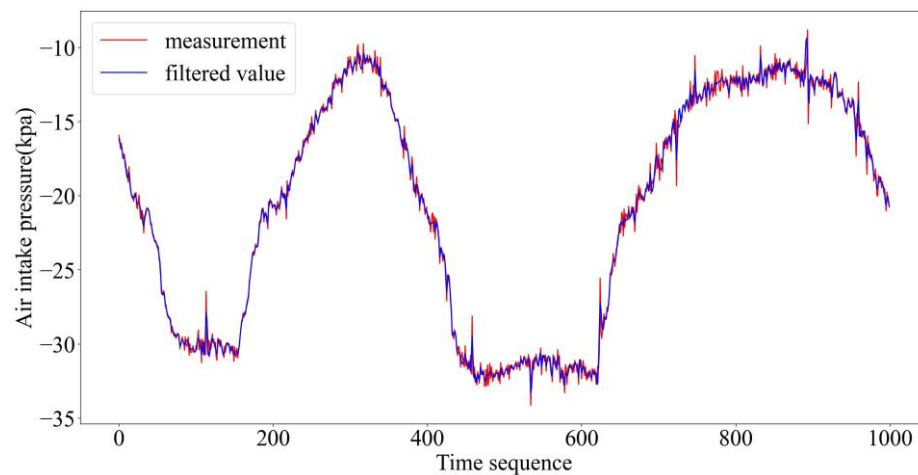
The magnitude difference between each feature parameter was too large, which affected the gradient update of network learning and reduced the prediction accuracy, so we adopted the z-score method to standardize the data, as shown in Formula (18):

$$x_z = \frac{x - \bar{x}}{S} \quad (18)$$

where  $S$  is Standard deviation of one-dimensional data and  $\bar{x}$  is mean value of the dimensional data.



**Figure 6.** Exhaust temperature after filtering.



**Figure 7.** Intake pressure after filtering.

## 5. Results and Discussion

### 5.1. Setting of the Experiments

We set the experiments on the dataset EDT, which is divided into training dataset, valid dataset, and testing dataset at a ratio of 7:2:1. The length of input sequence was set from 4 to 16 to evaluate the long sequence prediction ability of the network. As is shown in Figure 8, the training dataset included data collected from steady-state conditions and smooth acceleration and deceleration conditions. In Figure 9, the testing dataset consisted of a series of data which collected under engine rapidly changing conditions, such as sudden deceleration and rapid acceleration.

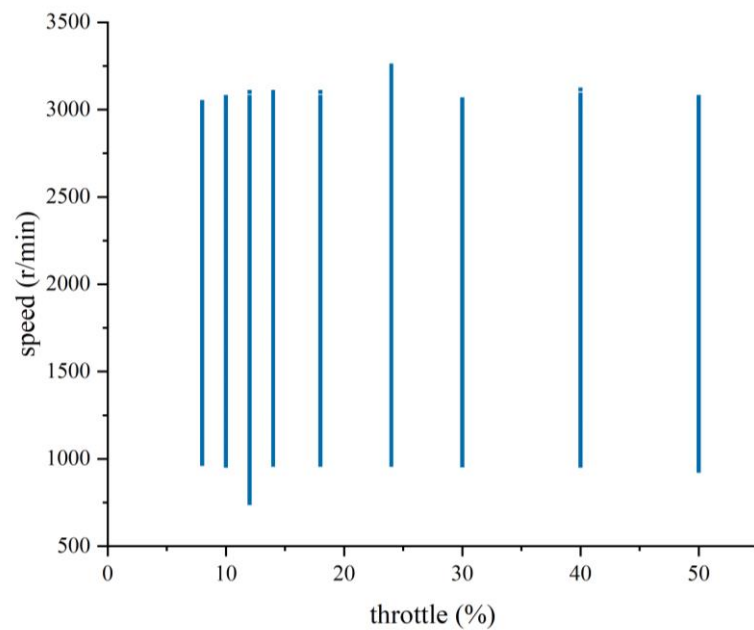
The experiments were conducted in the operating system Ubuntu 20.01 with AMD Core R9 CPU (12 core 3.8 GHz), 16 GB RAM memory and Nvidia GPU 3070. EGNN was implemented in Pytorch 1.7. In the experiments, we set the batch-size as 64 for training, validating, and testing. The optimizer we choose was the Adam optimizer that optimizes the parameters of the network to minimize the loss.

### 5.2. Baseline Models

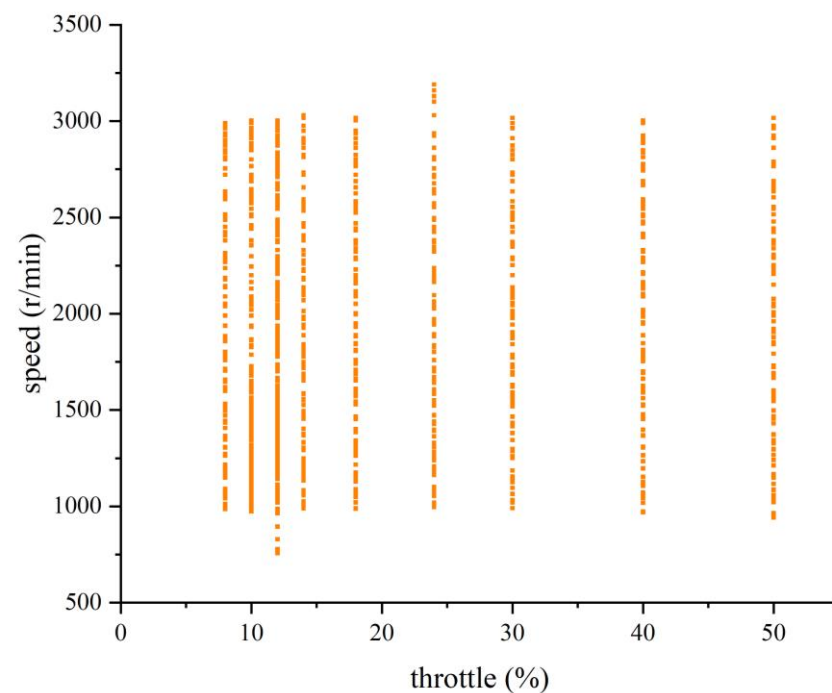
The EGNN was compared with the following five models.

Xgboost [35]: Xgboost, which is a traditional machine learning model for processing sequences.

LSTM [13]: A model based on RNN widely used in deep learning.



**Figure 8.** Engine operating envelop of the training set in terms of engine speed, and throttle valve percentage.



**Figure 9.** Engine operating envelop of the training set in terms of engine speed and throttle valve percentage.

GRU [36]: Another model changed from RNN that is applied on natural language processing.

Transformer [21]: Encoder-decoder architecture and the self-attention mechanism is first applied on it.

StemGNN [37]: StemGNN is a spatial-temporal GNN and based on graph structure automatically generated and attention mechanism, which can effectively mine the relationship between dynamic spatiotemporal features for time series prediction.

### 5.3. Performance Evaluation

Accuracy is a very important metric to measure the quality of NOx prediction models. At the same time, accurate prediction plays the important role in the engine control and calibration.

The mean absolute error (MAE), root mean square error (RMSE), precision and the coefficient of determination ( $R^2$ ) were used to evaluate the performance of our model on the EDT dataset. These metrics are represented by the Formulas (19)–(23), where  $y$  represents the true value,  $\hat{y}$  represents the predicted value of the model.  $n$  is the number of samples.

$$precision = \frac{1}{n} \sum (1 - \frac{|y_i - \hat{y}_i|}{y_i}) \tag{19}$$

$$RMSE = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2} \tag{20}$$

$$MAE = \frac{1}{n} \sum |y_i - \hat{y}_i| \tag{21}$$

$$R^2 = \frac{(n \sum y_i \hat{y}_i + \sum y_i \sum \hat{y}_i)}{[n \sum y_i^2 - (\sum y_i)^2][n \sum \hat{y}_i^2 - (\sum \hat{y}_i)^2]} \tag{22}$$

$$R = \sqrt{R^2} \tag{23}$$

### 5.4. Comparison of Graph Attention Mechanism Results

To investigate the validity of our proposed graph attention method, we performed a case study on two models.

Figures 10 and 11 show the result of graph attention mechanism for StemGNN and EGNN respectively. The data in each grid in the figure indicates the importance of the abscissa to the ordinate. For the completeness of the diagram, we performed attention score calculations for each variable and normalized by Formula (12). After normalization by Formula (13), each value shows the proportion of the influence of the variable on another variable, so the attention scores of the two graph neural networks can be compared.

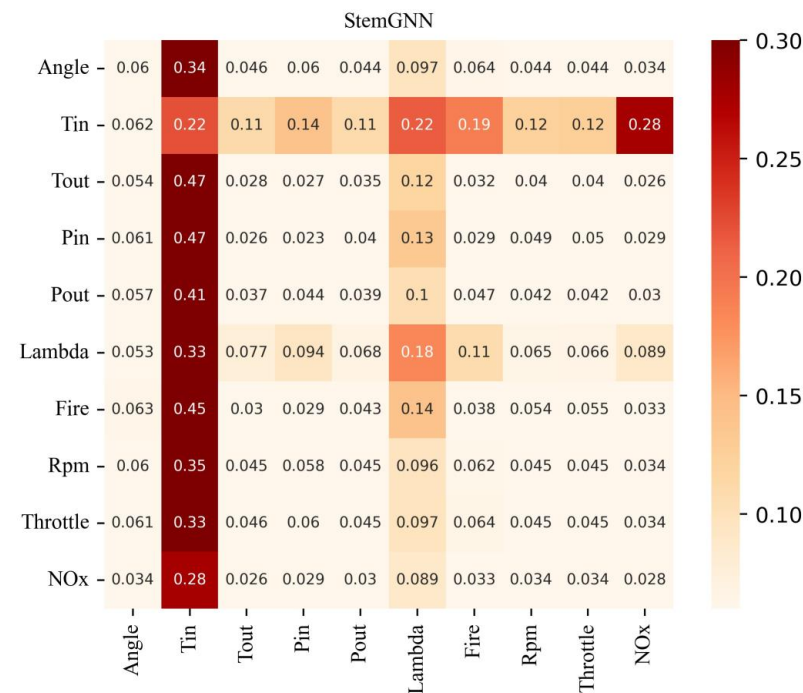


Figure 10. Graph attention result of StemGNN. The data in every grid represent the attention score of the variable to another variable.

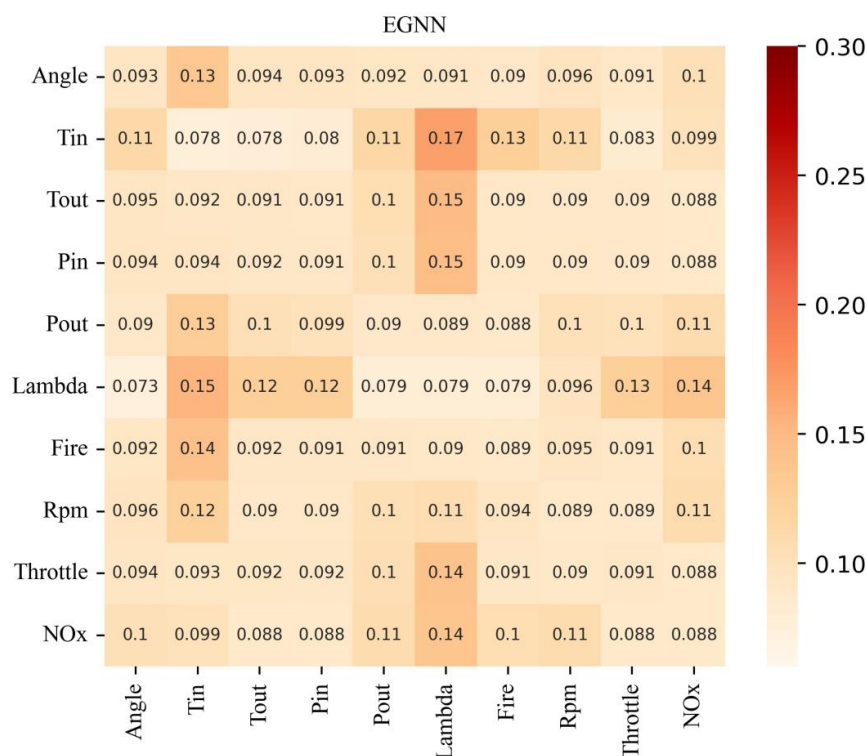


Figure 11. Graph attention result of EGNN. The data in every grid represent the attention score of the variable to another variable.

The variables are listed in Table 3. The StemGNN’s attention for NOx mainly focuses on intake temperature, and the attention values of other variables are very low, less than 0.04 while EGNN has a wider attention distribution for NOx where crankshaft angle, exhaust pressure, equivalence ratio, ignition angle and rotational speed have higher attention values. At the same time, crankshaft angle, exhaust pressure, equivalence ratio, ignition angle and rotational speed have a significant impact on emission values according to the emission characteristics of the engine. Therefore, our model not only obtains an outstanding forecasting performance, but also shows an advantage of interpretability.

Table 3. Implications of Variables in Figures 10 and 11.

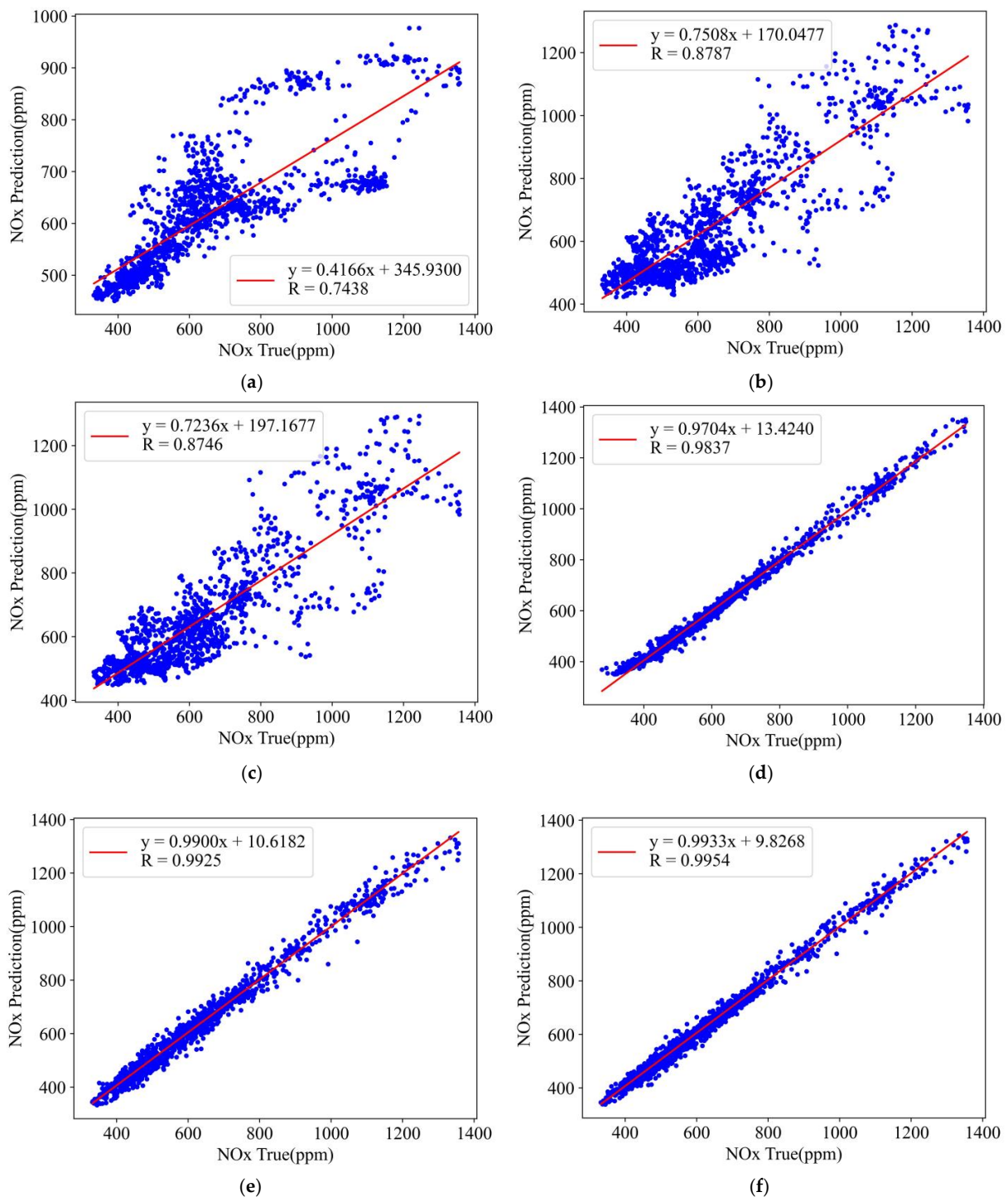
Variable Abbreviation	Implication
Angle	Crankshaft angle
Tin	Air intake temperature
Tout	Exhaust temperature
Pin	Air intake pressure
Pout	Exhaust pressure
Lambda	Equivalence ratio
Fire	Ignition angle
Rpm	Revolutions per minute
Throttle	Throttle valve percentage
NOx	Emissions

### 5.5. Prediction Results Analysis

In this section, we compare EGNN with the other five baseline methods based on the source dataset.

Figure 12 shows the predictions of the baseline models and EGNN under the R evaluation method whose horizontal and vertical coordinates corresponding to true values and the model prediction results. At the same time, the linear fitting function of the true values and predicted values are also shown in Figure 12. The closer the linear function

slope is to 1, the higher accuracy the model. The regression slopes were found to be 0.4166, 0.7553, 0.6963, 0.9802, 0.9900, 0.9933 for the six models, and EGNN had the highest slope, which means the best prediction ability. It was shown that the results of Xgboost, LSTM and GRU were more discrete compared to the  $Y = X$  function with the results of Transformer, StemGNN and EGNN. Traditional machine learning performed badly on our large sample datasets. Under the R index, StemGNN and Transformer were slightly worse than EGNN in single-step predictability, about 1.2% and 0.3%.



**Figure 12.** Linear regression and  $R^2$  results of the single time step under the random condition. (a) Xgboost; (b) LSTM; (c) GRU; (d) Transformer; (e) StemGNN; (f) EGNN.

As shown in Figure 13, EGNN achieved a good single time step prediction result. We selected the real values of random dynamic operating points and used six models to predict value for fitting. The predicted time step was one. It can be concluded that EGNN has a good prediction effect on the dataset even if under the random variation of speed and load and intensive changes in working conditions. When the engine operating conditions change rapidly, the first three models have extensive prediction distortion, and cannot accurately predict the change trend of the target value. Among them, although Xgboost can capture the change trend of emissions, it has a large deviation from the target value. Compared with the actual value, the prediction results of LSTM and GRU have obvious phase differences, which may be caused by the insufficient learning ability of RNN structure. The precisions of models are shown in Table 4. It is obvious that our method has the best precision at 97.3%, improving by 49.0, 13.0, 15.7, 0.6 and 0.6%, respectively, compared to the other five models. Comparing Figure 13d,f, the Transformer has a large error in some conditions while EGNN does not.

We used an emission sequence containing 24 time steps to predict the emission at different prediction length, including 4, 8, 12 and 16 time steps of NO<sub>x</sub> emission. From Table 5, it can be observed that: (1) our proposed model EGNN greatly improved prediction performance on the EDT dataset, and as the time step increased, the prediction error increased gradually and smoothly, which demonstrates the proposed sparse graph attention has great effects on enhancing the prediction capacity in the long time step emission forecast problem. (2) EGNN was superior to the GNN model and StemGNN at all tested time steps, which means the sparse graph self-attention mechanism can better mine nonlinear relationships among multi-variables. (3) The EGNN model shows significantly better ability and prediction results than recurrent neural networks such as LSTM and GRU. The proposed model had an RMSE decrease of 36.2% (at four steps), 23.1% (at eight steps), 18.7% (at twelve steps), and 17.3% (at sixteen steps). This indicates the graph structure had better prediction capacity than the RNN-based models. (4) Compared with Transformer based on encoder-decoder structure and self-attention mechanism, the EGNN also showed better prediction results.

Under the MAE and RMSE index, the prediction performance of our model was better than that of Transformer. As is shown in Figure 14, the advantages of EGNN are still obvious under the normalized MAE and RMSE evaluation methods, under the normalized MAE and RMSE evaluation methods.

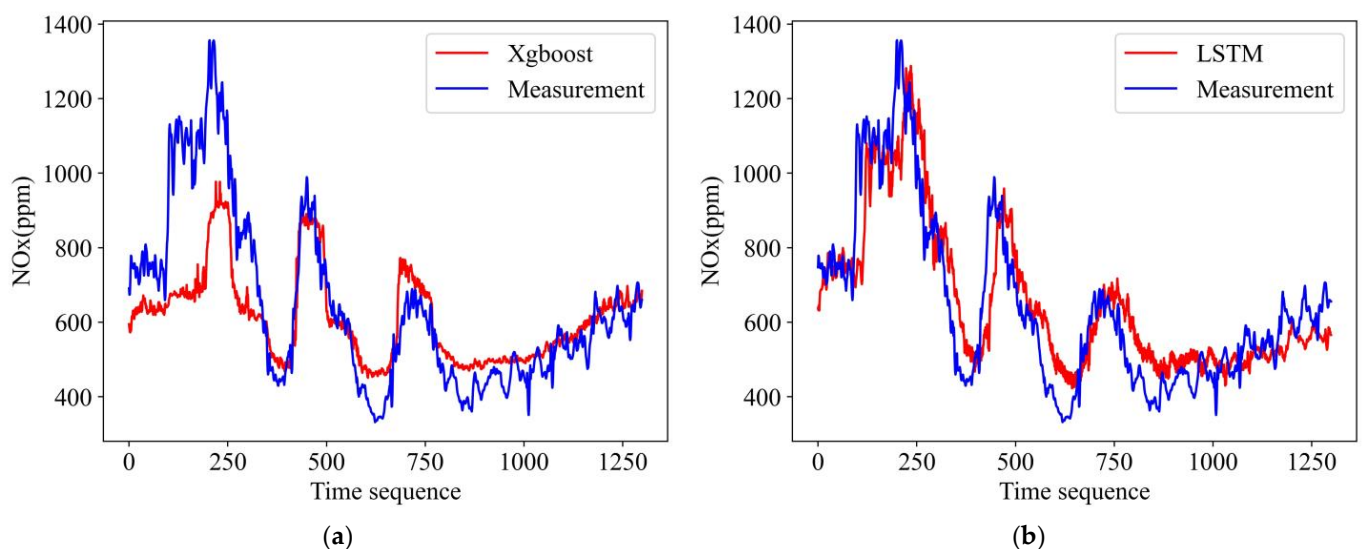
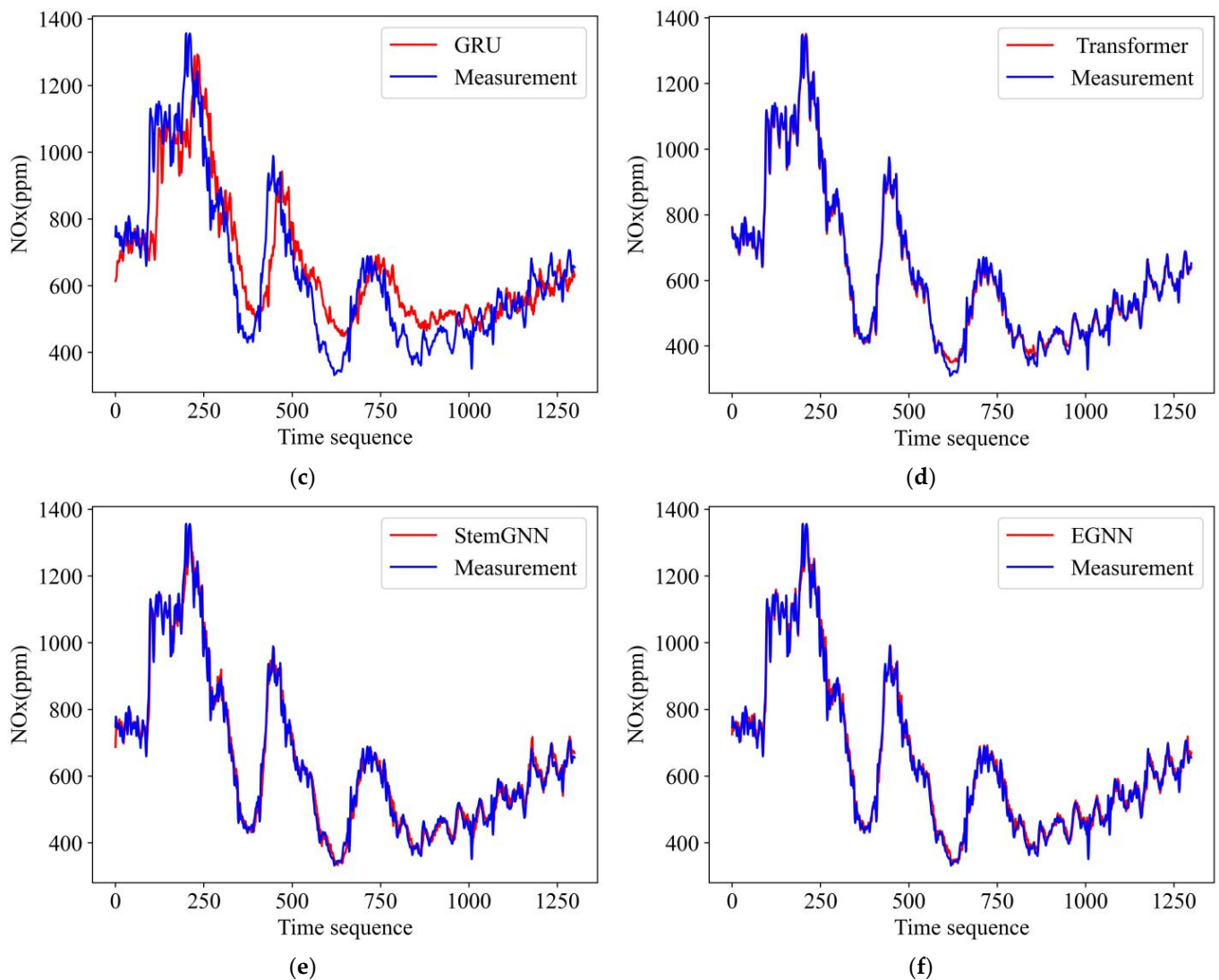


Figure 13. Cont.



**Figure 13.** Single time step prediction results of the random condition. (a) Xgboost; (b) LSTM; (c) GRU; (d) Transformer; (e) StemGNN; (f) EGNN.

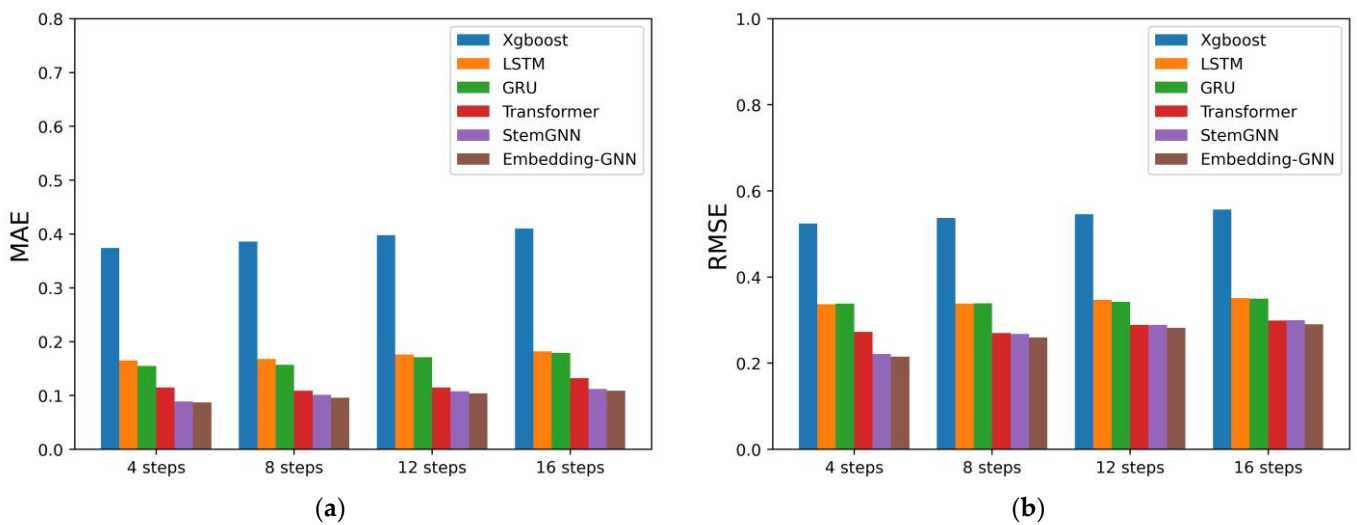
**Table 4.** Single time step prediction results of six models.

Metrics	Baseline Methods					
	Xgboost	LSTM	GRU	Transformer	StemGNN	EGNN
R	0.743	0.878	0.875	0.983	0.993	<b>0.995</b>
R <sup>2</sup>	0.553	0.771	0.741	0.968	0.985	<b>0.991</b>
Precision (%)	65.5	86.1	84.1	96.4	96.5	<b>97.3</b>
Linear slope	0.4166	0.7508	0.7236	0.9704	0.9900	<b>0.9933</b>

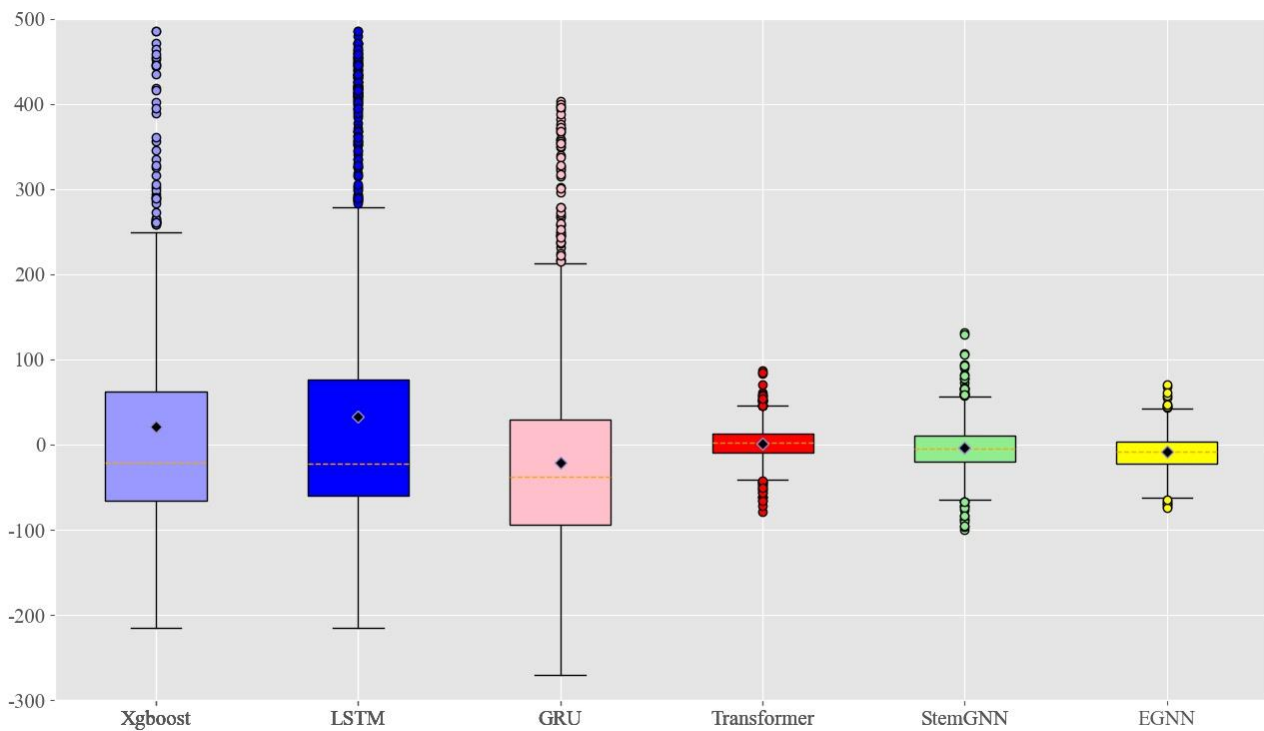
From Figure 15, showing the boxplot of prediction errors, it is easy to see that EGNN generates less higher prediction errors, which are marked as outliers in the plot, compared to the other five models. The greatest deviation of the predictions of the first three models close to 500 ppm. The deviations of the latter three were smaller. However, compared with EGNN, the prediction stability of the Transformer was worse, which is reflected in the larger number of deviation points of Transformer. It means our model has better stability. Combining Figure 15 and Table 5, we can observe that whether under a single time step or a longer time step, EGNN has better predictive ability.

**Table 5.** Prediction results of different time steps based on EGNN and the other five baseline models.

Time Steps	Metrics	Baseline Methods					
		Xgboost	LSTM	GRU	Transformer	StemGNN	EGNN
4	MAE (ppm)	41.09	17.31	17.03	9.066	7.016	<b>5.718</b>
	RMSE (ppm <sup>2</sup> )	98.21	50.24	49.80	22.41	18.14	<b>15.68</b>
8	MAE (ppm)	47.60	18.02	17.75	7.646	7.085	<b>6.356</b>
	RMSE (ppm <sup>2</sup> )	105.53	51.67	50.84	20.56	19.75	<b>18.66</b>
12	MAE (ppm)	50.19	18.76	18.99	8.105	7.343	<b>7.166</b>
	RMSE (ppm <sup>2</sup> )	111.71	52.16	52.71	21.73	21.73	<b>20.64</b>
16	MAE (ppm)	56.84	18.45	19.58	11.12	9.436	<b>7.842</b>
	RMSE (ppm <sup>2</sup> )	118.50	52.19	53.95	26.31	26.49	<b>23.55</b>



**Figure 14.** Normalized prediction results of different methods. (a) MAE. (b) RMSE.



**Figure 15.** Boxplot of single time step prediction errors. Outliers indicate large forecast errors.

## 6. Conclusions

We propose a neural network model composed of variable embedding, sparse graph attention and an EGNN block based on Fourier transform. In the present model, both steady-state and transient conditions were considered, and input variables were identified and utilized for improving the model performance. Several insightful results were achieved as follows:

- (1) Based on the feature embedding of Transformer, the Multidimensional Variable Embedding was developed. By assuming that each variable is a node, multidimensional variables was converted into a graph structure. Through the sparse self-attention mechanism technique, the conventional GNN model was optimized to achieve a better ability to capture interrelationships between variables.
- (2) The proposed graph neural network in this study provides a new route for effectively dealing with multi-dimensional variable correlation, high frequency variations and nonlinear NO<sub>x</sub> concentration time series. Compared with StemGNN, EGNN can focus on more variables in the calculation of correlation weights. Five of these variables received high weight scores in EGNN while StemGNN mainly focused on two variables. This suggests that EGNN can capture deeper interrelationships between variables.
- (3) Comparing the R<sup>2</sup> value of EGNN to other typical machine learning-based NO<sub>x</sub> predicting methodologies such as Xgboost, LSTM, GRU, the R<sup>2</sup> value of EGNN increased by 79.2, 28.5, and 33.8%, respectively, under the single timestep. However, the R<sup>2</sup> of EGNN only increased by 2.3% and 0.6% compared to Transformer and StemGNN.
- (4) On the other hand, the advantage of EGNN was demonstrated when the index of RSME was analyzed. Comparing the RSME of EGNN with Transformer and StemGNN, significant decreases of 21.4% and 8.8% could be achieved, which shows the unique long timestep prediction ability of the model developed in this study.
- (5) Increasingly stringent emission requirements generate a huge challenge for automotive engines. Being able to accurately predict the future emission values of the engine plays a key role in engine control. EGNN provides a possible route to reduce the cost of engine production and calibration processes, and provides good predictive support for engine control in the future.

The future research goal is to use different engines to study the transfer learning effect of the emission prediction network model to improve generalization.

**Author Contributions:** Data curation, Y.C. and X.M.; formal analysis, Q.N.; methodology, Y.C. and G.D.; project administration, D.L. and G.D.; resources, S.L. and X.H.; software, Y.C.; supervision, L.L.; validation, Y.C.; visualization, Y.C.; writing—original draft, Y.C. and C.L.; writing—review & editing, Y.C., Q.N., X.M. and X.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China under Grant 52176126, Longitudinal Project of Nanchang Automobile Innovation Research Institute TPDT202010-11 and Shanghai Science and Technology Plan Project 21ZR1464900.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

ANN	Artificial Neural Network	GNN	Graph Neural Network
BPNN	Back Propagation Neural Network	GRU	Gate Recurrent Unit
CEEMDAN	Complete Ensemble Empirical Mode Decomposition with Adaptive Noise	IDFT	Inverse Discrete Fourier Transform
CFD	Computational Fluid Dynamics	IGFT	Inverse Graph Fourier Transform
CO <sub>2</sub>	Carbon Dioxide	LSTM	Long and Short-term Memory
DNN	Deep Neural Network	MAE	Mean Absolute Error
DFT	Discrete Fourier Transform	NO <sub>x</sub>	Nitrogen Oxides
EGNN	Embedding-Graph-Neural-Network	RMSE	Root Mean Square Error
FS	Full Scale	RNN	Recurrent Neural Network
GFT	Graph Fourier Transform	StemGNN	Spectrogram Neural Network
GLU	Gated Linear Units	Xgboost	Extreme Gradient Boosting

## References

- Wang, H.; Ji, C.; Shi, C.; Ge, Y.; Meng, H.; Yang, J.; Chang, K.; Wang, S. Comparison and Evaluation of Advanced Machine Learning Methods for Performance and Emissions Prediction of a Gasoline Wankel Rotary Engine. *Energy* **2022**, *248*, 123611. [CrossRef]
- Ministry of Environmental Protection of the PRC. Emissions Standard of Air Pollutants for Thermal Power Plants. 2011. Available online: [http://en.cnki.com.cn/Article\\_en/CJFDTOTAL-YSC201012006.html](http://en.cnki.com.cn/Article_en/CJFDTOTAL-YSC201012006.html) (accessed on 2 July 2022).
- Ministry of Ecological and Environment of the PRC. Technical Guideline for the Development of National Air Pollutant Emission Standards. 2019. Available online: <http://english.mee.gov.cn/> (accessed on 2 July 2022).
- Bishop, J.D.K.; Stettler, M.E.J.; Molden, N.; Boies, A.M. Engine Maps of Fuel Use and Emissions from Transient Driving Cycles. *Appl. Energy* **2016**, *183*, 202–217. [CrossRef]
- Qi, K.; Feng, L.; Leng, X.; Du, B.; Long, W. Simulation of Quasi-Dimensional Combustion Model for Predicting Diesel Engine Performance. *Appl. Math. Model.* **2011**, *35*, 930–940. [CrossRef]
- Jung, D.; Assanis, D.N. Modeling of direct injection diesel engine emissions for a quasi-dimensional multi-zone spray model. *Int. J. Automot. Technol.* **2004**, *5*, 165–172.
- Rakopoulos, C.D.; Antonopoulos, K.A.; Rakopoulos, D.C. Development and Application of Multi-Zone Model for Combustion and Pollutants Formation in Direct Injection Diesel Engine Running with Vegetable Oil or Its Bio-Diesel. *Energy Convers. Manag.* **2007**, *48*, 1881–1901. [CrossRef]
- Hsieh, M.-F.; Wang, J. NO and NO<sub>2</sub> Concentration Modeling and Observer-Based Estimation Across a Diesel Engine Aftertreatment System. *J. Dyn. Syst. Meas. Control* **2011**, *133*, 041005. [CrossRef]
- Mellor, A.M.; Mello, J.P.; Duffy, K.P.; Easley, W.L.; Faulkner, J.C. Skeletal Mechanism for NO<sub>x</sub> Chemistry in Diesel Engines. *SAE Trans.* **1998**, *107*, 786–801.
- Liu, Y.; Midkiff, K.C.; Bell, S.R. Development of a Multizone Model for Direct Injection Diesel Combustion. *Int. J. Engine Res.* **2004**, *5*, 71–81. [CrossRef]
- Jung, D.; Assanis, D.N. Multi-Zone DI Diesel Spray Combustion Model for Cycle Simulation Studies of Engine Performance and Emissions. *SAE Trans.* **2001**, *110*, 1510–1532.
- Wang, X.; Liu, W.; Wang, Y.; Yang, G. A hybrid NO<sub>x</sub> emission prediction model based on CEEMDAN and AM-LSTM. *Fuel* **2022**, *310*, 122486. [CrossRef]
- Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
- Kiyas, H.; Sedat, M.; Selcuk, E.; Yasin, Ş. Developing a model for prediction of the combustion performance and emissions of a turboprop engine using the long short-term memory method. *Fuel* **2021**, *302*, 121202. [CrossRef]
- Akolaş, H.İ.; Kaleli, A.; Bakirci, K. Design and implementation of an autonomous EGR cooling system using deep neural network prediction to reduce NO<sub>x</sub> emission and fuel consumption of diesel engine. *Neural Comput. Applic.* **2021**, *33*, 1655–1670. [CrossRef]
- Golgiyaz, S.; Talu, M.F.; Onat, C. Artificial Neural Network Regression Model to Predict Flue Gas Temperature and Emissions with the Spectral Norm of Flame Image. *Fuel* **2019**, *255*, 115827. [CrossRef]
- Liukkonen, M.; Heikkinen, M.; Hiltunen, T.; Hälikkä, E.; Kuivalainen, R.; Hiltunen, Y. Artificial Neural Networks for Analysis of Process States in Fluidized Bed Combustion. *Energy* **2011**, *36*, 339–347. [CrossRef]
- Si, F.; Romero, C.E.; Yao, Z.; Schuster, E.; Xu, Z.; Morey, R.L.; Liebowitz, B.N. Optimization of Coal-Fired Boiler SCRs Based on Modified Support Vector Machine Models and Genetic Algorithms. *Fuel* **2009**, *88*, 806–816. [CrossRef]
- Zhao, F.; Ruan, Z.; Yue, Z.; Hung, D.L.S.; Som, S.; Xu, M. Time-Sequenced Flow Field Prediction in an Optical Spark-Ignition Direct-Injection Engine Using Bidirectional Recurrent Neural Network (Bi-RNN) with Long Short-Term Memory. *Appl. Therm. Eng.* **2020**, *173*, 115253. [CrossRef]
- Pascanu, R.; Mikolov, T.; Bengio, Y. On the Difficulty of Training Recurrent Neural Networks. In Proceedings of the 30th International Conference on Machine Learning, PMLR, Atlanta, GA, USA, 17–19 June 2013; pp. 1310–1318.

21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 5–9 December 2017; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30.
22. Kitaev, N.; Kaiser, Ł.; Levskaya, A. Reformer: The Efficient Transformer. *arXiv* **2020**, arXiv:2001.04451.
23. Li, S.; Jin, X.; Xuan, Y.; Zhou, X.; Chen, W.; Wang, Y.-X.; Yan, X. Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
24. Rae, J.W.; Potapenko, A.; Jayakumar, S.M.; Lillicrap, T.P. Compressive Transformers for Long-Range Sequence Modelling. *arXiv* **2019**, arXiv:1911.05507.
25. Goyal, P.; Ferrara, E. Graph Embedding Techniques, Applications, and Performance: A Survey. *Knowl. Based Syst.* **2018**, *151*, 78–94. [[CrossRef](#)]
26. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 11106–11115. [[CrossRef](#)]
27. Moradi, M.; Heinz, A.; Wagner, U.; Koch, T. Modeling the emissions of a gasoline engine during high-transient operation using machine learning approaches. *Int. J. Engine Res.* **2022**, *23*, 1708–1716. [[CrossRef](#)]
28. Mohammad, A.; Rezaei, R.; Hayduk, C.; Delebinski, T.; Shahpouri, S.; Shahbakhti, M. Physical-Oriented and Machine Learning-Based Emission Modeling in a Diesel Compression Ignition Engine: Dimensionality Reduction and Regression. *Int. J. Engine Res.* **2022**, 146808742110707. [[CrossRef](#)]
29. Norouzi, A.; Aliramezani, M.; Koch, C.R. A Correlation-Based Model Order Reduction Approach for a Diesel Engine NO<sub>x</sub> and Brake Mean Effective Pressure Dynamic Model Using Machine Learning. *Int. J. Engine Res.* **2021**, *22*, 2654–2672. [[CrossRef](#)]
30. Ma, C.; Yao, C.; Song, E.-Z.; Ding, S.-L. Prediction and Optimization of Dual-Fuel Marine Engine Emissions and Performance Using Combined ANN with PSO Algorithms. *Int. J. Engine Res.* **2022**, *23*, 560–576. [[CrossRef](#)]
31. Fang, X.; Papaioannou, N.; Leach, F.; Davy, M.H. On the Application of Artificial Neural Networks for the Prediction of NO<sub>x</sub> Emissions from a High-Speed Direct Injection Diesel Engine. *Int. J. Engine Res.* **2021**, *22*, 1808–1824. [[CrossRef](#)]
32. Fang, X.; Zhong, F.; Papaioannou, N.; Davy, M.H.; Leach, F.C. Artificial Neural Network (ANN) Assisted Prediction of Transient NO<sub>x</sub> Emissions from a High-Speed Direct Injection (HSDI) Diesel Engine. *Int. J. Engine Res.* **2022**, *23*, 1201–1212. [[CrossRef](#)]
33. Yu, Y.; Wang, Y.; Li, J.; Fu, M.; Shah, A.N.; He, C. A Novel Deep Learning Approach to Predict the Instantaneous NO<sub>x</sub> Emissions from Diesel Engine. *IEEE Access* **2021**, *9*, 11002–11013. [[CrossRef](#)]
34. Cao, J.; Li, Z.; Li, J. Financial Time Series Forecasting Model Based on CEEMDAN and LSTM. *Phys. A Stat. Mech. Its Appl.* **2019**, *519*, 127–139. [[CrossRef](#)]
35. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; Association for Computing Machinery: New York, NY, USA; pp. 785–794.
36. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
37. Cao, D.; Wang, Y.; Duan, J.; Zhang, C.; Zhu, X.; Huang, C.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; et al. Spectral Temporal Graph Neural Network for Multivariate Time-Series Forecasting. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 4 November–4 December 1999; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 17766–17778.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.