

Article

Short-Term Wind Power Forecasting Based on Spatio-Temporal Adaptive Graph Convolutional Recurrent Network

Shi Mo ^{1,*}, Xi Chen ¹, Zeyu Wang ² , Yuxiang Peng ¹, Bo Wang ¹ and Yixin Su ² 

¹ State Grid Hubei Economic Research Institute, Wuhan 430077, China; liumengoo@163.com (X.C.); pengyx22@foxmail.com (Y.P.); wbotton@163.com (B.W.)

² School of Automation, Wuhan University of Technology, Wuhan 430070, China; wzywzy@whut.edu.cn (Z.W.); suyixin@whut.edu.cn (Y.S.)

* Correspondence: ms15271812902@163.com

Abstract

The randomness and volatility of wind power pose significant challenges for short-term forecasting, requiring the model to capture both temporal dynamics and the spatial correlations among turbines. To address this issue, this paper proposes a Spatio-Temporal Adaptive Graph Convolutional Recurrent Network (STAGCRN). The proposed method dynamically constructs and updates the spatial relationship graph through node adaptive parameter learning (NAPL) and a data adaptive graph generation (DAGG) module, enabling more accurate modeling of spatio-temporal dependencies in wind power data. In addition, a spatio-temporal self-attention mechanism is introduced to enhance the model's ability to capture both short-term fluctuations and long-term temporal patterns. By stacking multiple spatio-temporal adaptive graph convolutional recurrent layers, the model is capable of extracting complex nonlinear characteristics in wind power sequences. Experimental results based on real wind farm data demonstrate that the proposed method achieves significantly improved prediction accuracy and robustness compared with existing approaches in short-term wind power forecasting tasks.

Keywords: wind power forecasting; spatio-temporal adaptive graph convolutional recurrent network; self-attention mechanism; spatio-temporal correlation

1. Introduction

Driven by the rising global energy demand and the escalating climate crisis, many countries are advancing energy system transformation toward low-carbon development. Wind energy, as one of the most prominent forms of renewable and clean power, plays a crucial role in contemporary power systems. However, the intrinsic intermittency and uncertainty of wind power output present significant challenges to maintaining grid stability, affecting generation efficiency and power quality, and in some cases even leading to wind curtailment. Therefore, how to achieve effective, rapid, and accurate short-term wind power forecasting has become an urgent problem to be addressed, as it is fundamental to enhancing the planning, dispatching, and operational performance of modern power systems.

At present, existing approaches for wind power forecasting are generally classified into three types: physical models, traditional statistical models, and deep learning-based techniques. Physical methods rely on meteorological data and terrain information, combined with the characteristic parameters of wind turbines, to establish models of wind



Academic Editor: Tonio Sant

Received: 7 November 2025

Revised: 9 December 2025

Accepted: 22 December 2025

Published: 24 December 2025

Copyright: © 2025 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

farms for power prediction [1]. Existing studies mainly focus on accurate numerical simulations of wind fields, which require consideration of detailed factors such as complex terrain and turbine operating conditions. Therefore, multi-field coupled wind field models must be adopted, resulting in high computational costs. Classical statistical forecasting methods do not account for the physical processes of wind speed variation; rather than relying on physical modeling, these approaches utilize historical observations to learn patterns and forecast future wind power generation. Feng et al. [2] employed a two-layer ensemble machine learning technique combined with deep feature selection to develop a data-driven multi-model wind power forecasting framework. However, this approach fails to capture sudden climatic changes or the nonlinear dynamic characteristics of wind speed. Sun et al. [3] extracted the main trend of wind power sequences through secondary decomposition, used the random forest algorithm to infer latent features, and employed clustering analysis to improve forecasting accuracy. However, this approach also increases model complexity and computational cost, which limits its suitability for real-time prediction and large-scale applications. Sheoran et al. [4] divided wind speed data into multiple sliding windows, within which independent Autoregressive Integrated Moving Average (ARIMA) models were constructed and fitted to achieve short-term wind speed forecasting. Nevertheless, this method relies on the assumption of time series stationarity and is highly sensitive to initial conditions, thereby restricting its flexibility and applicability in practice. In addition, statistical frameworks for wind power forecasting have increasingly integrated time-series models, such as Kalman filtering, Markov processes, and Bayesian inference, demonstrating their effectiveness in both domestic and global applications [5]. This kind of model is suitable for capturing linear correlations, but wind power data has nonlinear characteristics. As an emerging technology in the field of machine learning, wind power prediction methods based on deep learning are attracting more and more researchers' attention. Hu et al. [6] proposed an improved deep belief network hybrid prediction method, which realizes the processing of high-dimensional data, adaptive learning and the utilization of spatial correlation. Liu et al. [7] used the data division method and predicted through the LSTM optimized by genetic algorithms, and the system showed good generalization ability. Wu et al. [8] proposed a hybrid method based on deep-chain echo-state network and variable modal decomposition, which improves the problem that the existing multi-step prediction model ignores the time dependence between prediction steps. Although these methods have achieved promising results in wind power forecasting, they typically treat turbine observation points as independent units and perform convolution operations within a structured spatial framework. While this allows local spatial features to be captured, it often leads to the loss of global spatial dependencies. Such approaches can extract local spatial features but tend to lose global spatial information. In recent years, Graph Convolutional Networks (GCNs) have emerged as an effective tool for processing graph-structured data. Hou et al. [9] developed a Spatio-Temporal Fusion Module by integrating GCN and Convolutional Neural Network architectures, incorporating spatial dependencies into temporal input data (i.e., historical sequences). By leveraging the extracted spatio-temporal features, their model effectively captures long-term temporal dependencies. Chen et al. [10] designed a dynamic adjacency matrix to capture the global spatial weight distribution that changes over time between wind speed nodes, and combine the GRU and GCN to extract space-time characteristics to build a wind speed map network.

By effectively modeling the adjacency matrix, the spatial and dynamic characteristics of the wind farm turbine network can be fully captured, so as to improve the performance of the GNN model in the wind forecasting task. Traditional GCNs rely on predefined static adjacency matrices to describe the spatial correlation of wind output. However, due to the influence of geographical location and wind speed changes, this static representation

may not fully reflect the spatial relationship between turbine outputs. This restriction limits the application of GCN in extracting the spatial characteristics of wind power generation predictions. To address these issues, this paper proposes a short-term wind power forecasting method based on a STAGCRN. The main innovations are as follows:

- (1) A spatial graph construction strategy combining KNN topology with feature-based refinement is proposed, enabling the graph to better reflect dynamic spatial dependencies among turbines.
- (2) A multi-scale spatial–semantic correlation representation is developed by integrating turbine layout information with operational similarity features, providing a more informative topology for spatio-temporal modeling.
- (3) A spatio-temporal self-attention mechanism is incorporated to capture global dependencies and enhance spatial and temporal embeddings.
- (4) A unified STAGCRN architecture is constructed by stacking adaptive graph convolutional recurrent layers and a linear projection head, enabling efficient learning of spatio-temporal patterns for wind power forecasting.

2. Methodology

2.1. Problem Definition

The wind power output of a wind farm over time can be expressed as a matrix $\mathbf{X} \in \mathbb{R}^{T \times N}$, where T is the length of the time sequence and N denotes the number of turbines. Each column reflects the power generation record of an individual turbine. The task is to forecast the power output for the next τ time steps using the previous T observations. Accordingly, the forecasting problem can be described as learning a mapping function $f(\cdot)$ that infers future values from historical data:

$$f(\mathbf{X}_{1:T}; \theta) \rightarrow \mathbf{X}_{T+1:T+\tau} \quad (1)$$

where θ denotes the learnable parameters of the model.

To effectively describe the spatial correlations among turbines in a wind farm, the spatial structure is modeled as a graph $G = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V}$ denotes the set of nodes corresponding to turbines, $\mathcal{E}$ represents the connections between them, and $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix encoding spatial–semantic relationship weights among nodes. Based on this formulation, the forecasting task can be rewritten as:

$$f(\mathbf{X}_{1:T}, G; \theta) \rightarrow \mathbf{X}_{T+1:T+\tau} \quad (2)$$

2.2. Spatial–Semantic Distance Graph

In wind power time-series data, the correlations among turbines arise not only from their geographical locations but also from the similarity of their temporal fluctuation patterns. To jointly characterize these two types of dependencies, constructing a spatial–semantic distance graph that integrates spatial proximity with DTW-based semantic similarity.

Given two original power sequences $P = [a_1, a_2, \dots]$ and $Q = [b_1, b_2, \dots]$, their first-order differences are computed as $P' = [a_2 - a_1, a_3 - a_2, \dots]$, $Q' = [b_2 - b_1, b_3 - b_2, \dots]$. The DTW distances of the raw series and the differenced series are then combined through a weighted formulation:

$$D_{\text{comb}} = \alpha D_{\text{raw}} + (1 - \alpha) D_{\text{diff}} \quad (3)$$

where $D_{\text{raw}} = \text{DTW}(P, Q)$ and $D_{\text{diff}} = \text{DTW}(P', Q')$. In this study, both D_{raw} and D_{diff} are computed using the standard unconstrained DTW formulation, where the alignment path is normalized by its length to ensure comparability across sequences. The parameter

α was selected through grid search on the validation set to balance magnitude-based and trend-based similarity.

Spatial relationships among turbines are captured using a KNN-based graph constructed from normalized turbine coordinates, yielding the spatial graph G_s . The DTW-based combined distances described above are converted into the semantic similarity graph G_t .

$$G_{st}(i, j) = \beta G_s(i, j) + (1 - \beta) G_t(i, j) \quad (4)$$

where $\beta \in [0, 1]$ balances spatial proximity and semantic similarity. All edge weights are subsequently normalized to form a valid adjacency matrix for downstream graph learning.

The resulting spatial–semantic distance graph G_{st} thus provides a richer structural representation that jointly incorporates spatial layout and temporal variation patterns of turbines.

2.3. Node Adaptive Parameter Learning

In the field of wind power forecasting, GCNs have been widely applied to capture the spatial dependencies among turbines. Typically, spectral-domain computation is adopted for graph convolution operations [11], where first-order Chebyshev polynomial approximation is used to simplify the Laplacian computation. To further generalize the convolution operation to high-dimensional graph modeling [12], the GCN layer can be expressed as:

$$Z = \left(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) X_i \Theta + b \quad (5)$$

where $Z \in \mathbb{R}^{N \times F}$ is the output feature matrix of the GCN layer, I_N denotes the N -dimensional identity matrix, and $X_i \in \mathbb{R}^{N \times C}$ represents the input feature set, where N is the number of wind turbines and C is the dimension of input features. D is the degree matrix associated with the graph structure. The matrices $\Theta \in \mathbb{R}^{C \times F}$ and $b \in \mathbb{R}^F$ are trainable parameters for weights and biases, respectively. The value F corresponds to the output dimensionality used for the τ -step forecasting task.

In Equation (5), traditional GCNs employ shared parameters Θ and b for all turbine nodes, which allows efficient learning of common structural patterns while reducing model complexity. Nevertheless, the shared-parameter strategy hampers the model's capability to characterize the non-uniform interaction structures of turbine nodes. To address this issue, a unique parameter space is assigned to each turbine node through NAPL. The learnable parameters $\Theta \in \mathbb{R}^{C \times F}$ are decomposed into two embedding matrices: a node embedding matrix $E \in \mathbb{R}^{N \times d}$ and a node weight pool $\bar{W} \in \mathbb{R}^{d \times C \times F}$, where the embedding dimension d is much smaller than N . By factorizing in this manner, the parameters and bias terms can be represented as:

$$\begin{cases} \Theta = E \bar{W} \\ b = E \bar{b} \end{cases} \quad (6)$$

where $\bar{b} \in \mathbb{R}^{d \times F}$ is the bias matrix.

Thus, the NAPL-GCN module can be reformulated as:

$$Z = \left(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) X_i E \bar{W} + E \bar{b} \quad (7)$$

2.4. Data-Adaptive Graph Generation

Most GCN-based wind power forecasting methods rely on predefined adjacency matrices to capture spatial dependencies. These matrices are typically constructed based on geographic distances or empirical similarity metrics. Although intuitive, such approaches often fail to comprehensively represent spatial correlations and heavily depend on expert

knowledge. To overcome this limitation, the DAGG module dynamically generates adjacency structures within neural networks based on the data itself, thereby better capturing the spatio-temporal correlations of wind power data.

(1) Node embedding initialization. Each node feature in the DAGG module is projected into a low-dimensional embedding space through a learnable embedding dictionary $E_A \in \mathbb{R}^{N \times d_e}$. In this matrix, each row corresponds to the learned representation of a turbine node, and d_e specifies the size of the embedding dimension.

(2) Node similarity computation. By quantifying the similarity between node embedding vectors, the potential relationships among nodes can be inferred. The similarity is usually computed via the dot product of embedding vectors, yielding a new pairwise similarity matrix represented as $E_A E_A^T$.

(3) Generation of adaptive adjacency matrix. An adaptive adjacency matrix is constructed from the learnable node embeddings $E_A \in \mathbb{R}^{N \times d}$. A similarity matrix is obtained as $S = \text{ReLU}(E_A E_A^T) \in \mathbb{R}^{N \times N}$, and a row-wise Softmax is applied to produce the adaptive adjacency.

$$A_{\text{dagg}}(i, j) = \text{Softmax}_{\text{row}} S_{ij} = \frac{e(S_{ij})}{\sum_{k=1}^N e(S_{ik})} \quad (8)$$

where $\text{Softmax}_{\text{row}}(\cdot)$ denotes normalization along each row so that $\sum_j A_{\text{dagg}}(i, j) = 1$. No explicit symmetry constraint is imposed, allowing the model to learn directed spatial dependencies. Self-loops are incorporated in the subsequent graph convolution through the identity matrix I_N .

Consequently, the resulting graph convolution based on the adaptive adjacency can be written as

$$Z = (I_N + A_{\text{dagg}}) X_i \Theta + b \quad (9)$$

where X_i is the node feature matrix at time step i , and Θ and b denote the learnable weights and bias, respectively.

When integrating the learned adaptive adjacency A_{dagg} with the spatial-semantic graph G_{st} , the final convolutional formulation is given by

$$Z = \left(I_N + A_{\text{dagg}} + D_{st}^{-\frac{1}{2}} G_{st} D_{st}^{-\frac{1}{2}} \right) X_i \Theta + b \quad (10)$$

where D_{st} is the degree matrix of G_{st} , defined as $D_{st,ii} = \sum_j G_{st}(i, j)$.

In Equation (10), D_{st} denotes the degree matrix associated with G_{st} . By integrating similarity-based node embeddings with the DTW-derived spatial-semantic graph, this formulation enhances the model's ability to capture the underlying spatio-temporal dependencies in wind power data.

2.5. Temporal Self-Attention Module

To enhance temporal dependency modeling prior to recurrent graph learning, a lightweight temporal self-attention module is introduced and applied to the input sequence $X \in \mathbb{R}^{T \times N \times 1}$. For each turbine, the sequence is reshaped into $\tilde{X} \in \mathbb{R}^{(BN) \times T \times F}$, where $F = 1$ denotes the feature dimension used in the temporal encoder.

A single-head scaled dot-product attention mechanism is adopted. The queries, keys, and values are produced through linear projections:

$$Q = \tilde{X} W_Q, \quad K = \tilde{X} W_K, \quad V = \tilde{X} W_V \quad (11)$$

where $W_Q, W_K, W_V \in \mathbb{R}^{F \times F}$ are learnable parameters. The attention weights are computed as

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{F}}\right) \quad (12)$$

The output of the attention module is subsequently mapped back to the original tensor layout and combined with the input via a residual connection $X_{\text{att}} = X + \text{reshape}(\text{Attention}(Q, K, V)V)$, which stabilizes optimization and preserves local temporal patterns.

The module employs a single-head structure without positional encoding or normalization layers, reflecting the short forecasting horizon and ensuring compatibility with the subsequent STAGCRN recurrent update. The resulting tensor X_{att} serves as the refined temporal input to the STAGCRN layers.

2.6. Spatio-Temporal Adaptive Graph Convolutional Recurrent Network

The GRU updates and resets hidden states through its gating mechanism, allowing it to effectively model temporal information and retain long-range sequence patterns. However, conventional graph convolution approaches are often limited in capturing spatial interactions among turbines. To address this issue, we develop a spatio-temporal adaptive graph convolutional recurrent module that combines spatio-temporal embeddings, node-adaptive parameter learning, dynamic graph construction, and GRU-based temporal modeling, enabling deeper representation of spatio-temporal relationships within wind farms. The overall architecture of the proposed STAGCRN framework is summarized in Algorithm 1.

Algorithm 1: STAGCRN

Input: Sequence $X_{1:T}$, horizon τ , KNN size k , weights α, β .

Output: Forecast $\hat{X}_{T+1:T+\tau}$.

1. Spatial-Semantic Graph

Compute raw and differenced DTW distances;

Fuse distances (α) to form semantic graph G_t ;

Build spatial graph G_s using KNN (k neighbors);

Fuse G_s and G_t (β) to obtain G_{st} .

2. DAGG

Initialize node embeddings;

Compute similarity matrix and apply row-wise Softmax to obtain A_{dagg} .

3. NAPL

Generate node-specific weights and biases from node embeddings.

4. Recurrent Spatio-Temporal Modeling

Initialize hidden state H_0 ;

for $t = 1$ **to** T **do**

 Apply temporal self-attention to X_t ;

 Perform graph convolution using A_{dagg} and G_{st} ;

 Update reset gate, update gate, and hidden state in the STAGCRN cell.

5. Prediction

Use H_T to generate $\hat{X}_{T+1:T+\tau}$.

As illustrated in Figure 1, the proposed network consists of multiple submodules. The initial unit receives the output F_{out} from the spatio-temporal self-attention module,

while subsequent units take the output of the previous cell as their input. In this work, the NAPL-GCN module replaces the multilayer perceptron (MLP) components within the standard GRU, forming the adaptive graph unit G^* . The network operations can be formulated as follows:

$$\begin{cases} r_t = \sigma(\theta([x_t \oplus H_{t-1}, Z])EW_r + Eb_r) \\ u_t = \sigma(\theta([x_t \oplus H_{t-1}, Z])EW_u + Eb_u) \\ \hat{h}_t = \tanh(\theta([x_t \oplus r_t \odot H_{t-1}, Z])EW_c + Eb_c) \\ H_t = u_t \odot H_{t-1} + (1 - u_t) \odot \hat{h}_t \end{cases} \quad (13)$$

where r_t and u_t represent the reset gate and update gate of the GRU, which regulate how much past information is preserved in the hidden state. The variable $\hat{h}_t$ is the candidate hidden state, and x_t and H_t are the input and output at the current time step. The activation functions $\sigma(\cdot)$ and $\tanh(\cdot)$ correspond to the sigmoid and hyperbolic tangent functions, respectively. The operator $\oplus$ indicates feature concatenation, $\odot$ denotes element-wise multiplication, and $\theta(\cdot)$ represents the dynamic adaptive graph generation operation. E , W_r , W_u , W_c and b_r , b_u , b_c are the learnable parameters of the network.

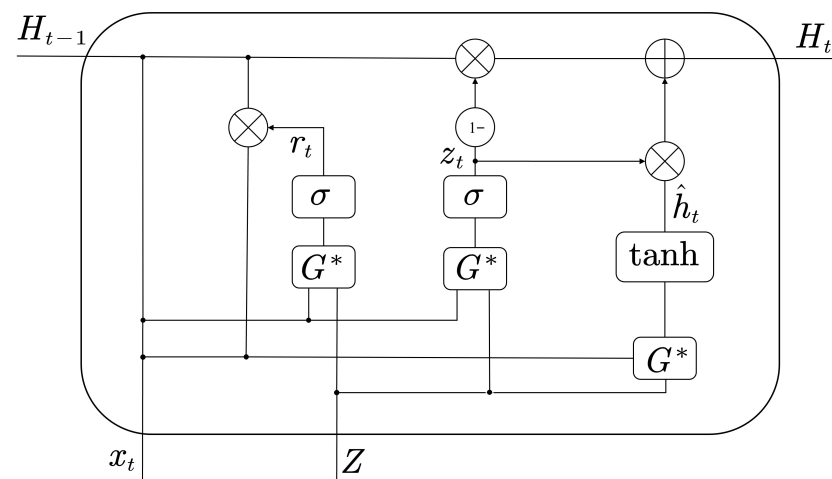


Figure 1. Spatio-Temporal Adaptive Graph Convolutional Recurrent Unit.

3. Experiments and Analysis

In this study, two types of experimental environments were used for model training:

(1) On a Windows platform, the experiments were conducted using the PyTorch framework (version 1.13.1) with CUDA 11.7, running on an NVIDIA GeForce RTX 4060 Laptop GPU with an Intel Core i9-14900HX processor and 16 GB RAM.

(2) On a cloud server platform with the Linux operating system, the experiments were conducted using the PyTorch framework on an NVIDIA RTX 4090D GPU and an 18-vCPU AMD EPYC 9754 processor, equipped with 60 GB RAM.

3.1. Case Analysis

The experiment utilized a real-world wind farm dataset derived from the Baidu KDD Cup 2022 (SDWPF), which contains comprehensive operational data from 200 wind turbines (monitoring points) [13]. The sampling cycle is 1 h, from 1 September 2010 to 31 August 2011, and a total of 8760 sets of wind power data were recorded. Missing values in the dataset were handled through a two-stage imputation strategy: linear interpolation was first applied to fill short gaps, followed by forward and backward filling to ensure temporal continuity. After imputation, Min-Max normalization was used to scale all input

variables into the range $[0, 1]$, which accelerates gradient descent convergence and stabilizes model training, as shown in Equation (14).

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (14)$$

where x is the raw feature value, $x_{\max}$ and $x_{\min}$ denote its maximum and minimum values, respectively, and x' is the value after normalization.

The dataset was partitioned into training, validation, and testing sets in a 6:2:2 ratio. The batch size was fixed to 32. The input sequence length was 48, while the prediction horizon was 1, indicating that 48 h of historical wind power data were utilized to forecast the power output for the subsequent hour. The model was trained for up to 100 epochs using the Adam optimizer with a learning rate of 0.001. To avoid model overfitting, an Early Stopping mechanism was applied, where the training process was halted if the validation metric failed to improve for 10 consecutive epochs. The model was optimized using the mean squared error (MSE) loss function, and all learnable node embeddings, including the NAPL embedding matrix E and the DAGG embedding matrix E_A , were initialized using Xavier normal initialization. Since the variation of MAE, MAPE, and RMSE across different hidden dimensions was relatively small and did not affect the final model choice, the hidden size was fixed at 72 in all experiments.

3.2. Model Evaluation Metrics

To assess the effectiveness of the proposed model, three widely used evaluation metrics were employed: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). These indicators quantify the discrepancy between the predicted outputs and the actual wind power measurements, thereby reflecting the model's prediction accuracy as well as the relative magnitude of forecasting errors, as defined in Equation (15).

$$\begin{cases} \text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \\ \text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{\hat{y}_i - y_i}{y_i} \right| \times 100 \\ \text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2} \end{cases} \quad (15)$$

where $\hat{y}_i$ and y_i denote the predicted and actual values of the i -th sample, respectively, and n is the total number of samples in the testing set.

3.3. Wind Power Forecasting Comparison Experiments

To evaluate the effectiveness of the proposed STAGCRN model, a comparative study was conducted against several representative sequence prediction methods, including RNN, LSTM, GRU, and Transformer-based architectures. In this experiment, the number of hidden layer neurons in RNN, LSTM and GRU is set to 72. For the Transformer-based model, the encoder consists of stacked multi-head self-attention layers and position-wise feed-forward networks, following the standard architecture used in long-term time-series forecasting. The parameter configuration of the STAGCRN network is shown in Table 1.

As shown in Table 2, STAGCRN achieves the best overall forecasting performance, obtaining the lowest MAE 0.0710, RMSE 0.1098, and MAPE 13.56%. These results indicate that jointly modeling spatial dependencies through adaptive graph learning and temporal dynamics through recurrent structures yields substantial gains in predictive accuracy.

Table 1. Parameter configuration of the STAGCRN network.

Parameter	Configuration
Num nodes	200
Hidden dim	72
Embed dim	20
Num layers	2

Table 2. Comparison of different models on wind power forecasting performance.

Model	MAE	RMSE	MAPE	Time(s)
RNN	0.2893	0.5326	75.95%	16.88
LSTM	0.1017	0.3129	46.40%	23.22
GRU	0.0998	0.1861	37.41%	16.99
Transformer	0.0815	0.1180	28.72%	21.64
STAGCRN	0.0710	0.1098	13.56%	643.16

The recurrent baselines (RNN, LSTM, and GRU) demonstrate relatively low computational cost, with inference times ranging from 16.88 s to 23.22 s. Nevertheless, their predictive performance remains limited due to the lack of explicit spatial dependency modeling. For instance, RNN yields an RMSE of 0.5326 and a MAPE of 75.95%, indicating substantial forecasting errors. Although GRU achieves lower error than RNN and LSTM, its MAE and RMSE still fall short of those achieved by STAGCRN, highlighting the insufficiency of recurrent architectures for capturing complex spatiotemporal dynamics in wind power series. Transformer-based models provide improved performance relative to recurrent baselines, benefiting from their capability to model long-range temporal dependencies. However, the standard Transformer records an RMSE of 0.1180, which remains higher than that of STAGCRN 0.1098. These results indicate that global self-attention alone is insufficient for fully capturing the localized spatial interactions inherent in wind turbine networks.

In terms of computational burden, STAGCRN requires 643.16 s for inference—substantially higher than recurrent and Transformer-based models. This increased overhead is attributed to the integration of DAGG for adaptive graph construction, NAPL for node-wise parameter learning, and a temporal attention mechanism, all of which introduce additional model complexity. Despite this, the accuracy improvements achieved by STAGCRN demonstrate a favorable trade-off, suggesting that the enhanced modeling of spatiotemporal dependencies justifies the computational cost in practical short-term wind power forecasting applications.

To further evaluate the effectiveness of the proposed STAGCRN model, several representative forecasting architectures, including Informer [14], Autoformer [15], and Fedformer [16], were selected for comparison. As shown in Table 3, STAGCRN achieves the best overall performance across all metrics, obtaining the lowest MAE, RMSE, and MAPE. These results indicate that the proposed model can more effectively capture the complex spatio-temporal dependencies inherent in wind power data.

Table 3. Comparison of wind power forecasting methods.

Model	MAE	RMSE	MAPE
Informer	0.0914	0.1206	35.63%
Autoformer	0.0833	0.1174	28.85%
Fedformer	0.0757	0.1113	17.29%
STAGCRN	0.0710	0.1098	13.56%

Among the Transformer-based models, Fedformer provides the most competitive results, with MAE 0.0757 and RMSE 0.1113, but its accuracy still remains below that of STAGCRN. Autoformer shows comparable performance, with RMSE values of 0.1174, respectively. Informer produces the weakest results among all compared methods, with MAE 0.0914 and RMSE 0.1206, indicating its limited capability in handling the short-term variability of wind power data. Overall, these comparison results show that STAGCRN achieves consistently better forecasting accuracy than both recurrent and Transformer-based baselines. This improvement can be attributed to the incorporation of DAGG, NAPL, and the attention mechanism, which together enable more effective extraction of spatial and temporal dependencies.

To evaluate the robustness of the proposed STAGCRN model under different forecasting horizons, multi-step prediction experiments were conducted for 2, 4, and 8-step settings. As shown in Table 4, STAGCRN achieves the lowest MAE, RMSE, and MAPE across all horizons. For the 2-step forecast, the model attains MAE 0.0732 and RMSE 0.1203, outperforming both recurrent models (RNN, LSTM, GRU) and Transformer-based architectures. Although prediction errors increase for all methods as the horizon becomes longer, STAGCRN maintains comparatively stable performance, achieving MAE 0.0961 at 4 steps and 0.1584 at 8 steps. Transformer-based baselines exhibit varying degrees of performance degradation with increasing horizon lengths. Fedformer produces the most competitive results among them, yet its accuracy remains below that of STAGCRN at all horizons. Informer and the standard Transformer show larger increases in error, while Autoformer provides relatively moderate performance. Overall, the multi-horizon results demonstrate that the adaptive spatio-temporal modeling mechanism employed by STAGCRN enables more reliable forecasting across both short and extended prediction horizons.

Table 4. Multi-horizon forecasting performance comparison.

Model	2-Step			4-Step			8-Step		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
STAGCRN	0.0732	0.1203	18.61%	0.0961	0.1349	39.64%	0.1584	0.1945	48.96%
RNN	0.1244	0.1637	35.34%	0.1706	0.2142	51.93%	0.2048	0.2535	54.28%
LSTM	0.1114	0.1493	28.42%	0.1501	0.1909	48.58%	0.1977	0.2424	53.36%
GRU	0.1027	0.1422	27.45%	0.1300	0.1684	43.78%	0.1739	0.2108	51.17%
Transformer	0.1390	0.1755	37.86%	0.1749	0.2075	50.85%	0.1978	0.2277	52.29%
Informer	0.1105	0.1512	31.32%	0.1387	0.1785	46.19%	0.1749	0.2091	51.12%
Autoformer	0.1001	0.1430	38.58%	0.1376	0.1788	45.76%	0.1836	0.2177	52.81%
Fedformer	0.1028	0.1517	29.39%	0.1249	0.1685	40.49%	0.1815	0.2151	52.35%

To further verify the prediction accuracy of the proposed STAGCRN model, six representative wind turbines with node indexes of 1, 40, 80, 120, 160 and 200 were selected. Figure 2 shows the comparison between the predicted wind power value of these wind turbines and the actual wind power value. As shown in Figure 2, the model performs well in fitting the overall trend, especially in the stable period of wind power change or in low-power areas, and the predicted value coincides well with the actual value. This shows that the model can effectively capture the time characteristics of the wind power sequence and make accurate and stable predictions of the long-term wind power sequence. In addition, thanks to the extraction of space-time characteristics, the fluctuation between the predicted wind power curve and the actual wind power curve is extremely small, which further verifies the robustness and adaptability of the proposed model.

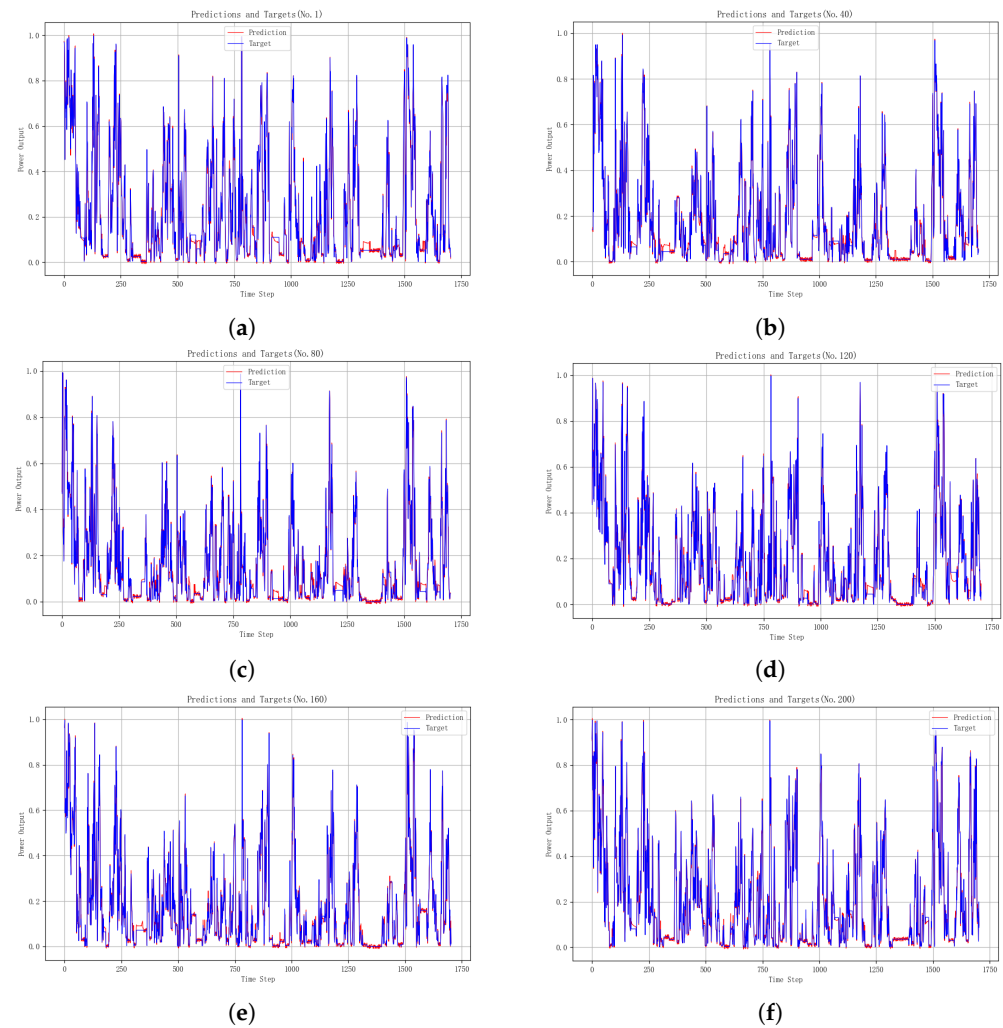


Figure 2. Wind turbine power prediction results: (a) Turbine No. 1; (b) Turbine No. 40; (c) Turbine No. 80; (d) Turbine No. 120; (e) Turbine No. 160; (f) Turbine No. 200.

3.4. Ablation Study

To further evaluate the contribution of each core component in the proposed STAGCRN model, three ablation variants were constructed:

- (1) STAGCRN_wo_DAGG: the data-adaptive graph generation module is removed, and only the static KNN-based adjacency is used;
- (2) STAGCRN_wo_NAPL: node-adaptive parameter learning is disabled, and all nodes share the same graph convolution parameters;
- (3) STAGCRN_wo_Attn: the temporal self-attention mechanism is removed, and the input sequence is fed directly into the recurrent layers.

Compared with the full STAGCRN model, the performance degradation observed in the ablation variants demonstrates the effectiveness of each core component, as shown in Table 5. Removing the DAGG module increases MAE, RMSE, and MAPE by 16.48%, 6.74%, and 112%, respectively, indicating that dynamically learned adjacency plays an important role in modeling turbine-level spatial relationships. Removing the NAPL module leads to the largest degradation, with MAE, RMSE, and MAPE increasing by 21.13%, 12.10%, and 137%, confirming that node-adaptive parameter learning is crucial for capturing heterogeneous turbine behavior. Disabling the self-attention mechanism also weakens the model, as MAE, RMSE, and MAPE increase by 11.69%, 6.83%, and 32.2%, respectively, demonstrating that attention-based feature enhancement contributes to modeling temporal

variability. These findings collectively validate that DAGG, NAPL, and self-attention form essential components of the proposed STAGCRN framework and jointly contribute to its forecasting accuracy.

Table 5. Ablation results of the proposed STAGCRN model.

Model	MAE	RMSE	MAPE
STAGCRN	0.0710	0.1098	13.56%
STAGCRN_wo_DAGG	0.0827	0.1172	28.75%
STAGCRN_wo_NAPL	0.0860	0.1231	32.12%
STAGCRN_wo_Attn	0.0793	0.1173	17.93%

4. Conclusions

This paper proposes a STAGCRN to solve the challenge of short-term wind power forecasting. The model constructs a static spatial graph by integrating the spatial correlations among wind turbine units and the temporal variation patterns of wind power output. On this basis, a data-driven adaptive graph learning mechanism is introduced to dynamically refine the adjacency structure, enabling more accurate modeling of evolving spatio-temporal dependencies. Moreover, the incorporation of a gated recurrent unit enhances temporal feature extraction and strengthens sequence modeling capability. Extensive experiments on real wind farm datasets demonstrate that the proposed model achieves superior forecasting performance compared with several widely used benchmark methods. Furthermore, the ablation studies validate the necessity and effectiveness of each core component in the framework.

In the future work, we will integrate meteorological variables and atmospheric conditions to further investigate their influence on spatio-temporal patterns in wind power generation. Additionally, improving the model's generalization across multiple wind farm scenarios and enhancing its robustness under extreme weather conditions will be key directions for subsequent research.

Author Contributions: Conceptualization, S.M. and Y.S.; methodology, Z.W.; software, Z.W.; validation, Z.W., Y.P. and B.W.; formal analysis, Z.W.; investigation, X.C.; resources, S.M.; data curation, Z.W.; writing—original draft preparation, Z.W.; writing—review and editing, S.M. and Y.S.; visualization, Z.W.; supervision, S.M. and Y.S.; project administration, S.M.; funding acquisition, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the State Grid Hubei Electric Power Company Science and Technology Project (Grant No. 52153825000A).

Data Availability Statement: The wind power dataset used in this study is publicly available from the Baidu KDD Cup 2022 Spatial Dynamic Wind Power Forecasting Challenge (SDWPF) at <https://aistudio.baidu.com/aistudio/competition/detail/152/0/introduction>, accessed on 12 October 2024. No new data were created or analyzed in this study beyond this publicly available dataset.

Conflicts of Interest: Authors Shi Mo, Xi Chen, Yuxiang Peng and Bo Wang were employed by the State Grid Hubei Economic Research Institute. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

STAGCRN	Spatio-Temporal Adaptive Graph Convolutional Recurrent Network
NAPL	Node Adaptive Parameter Learning
DAGG	Data-Adaptive Graph Generation
DTW	Dynamic Time Warping
KNN	k-Nearest Neighbors
GCN	Graph Convolutional Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory Network
GRU	Gated Recurrent Unit
Informer	Transformer architecture with ProbSparse self-attention for long-sequence forecasting
Autoformer	Transformer model incorporating series decomposition and auto-correlation mechanisms
Fedformer	Frequency-domain enhanced Transformer with decomposition-based forecasting structure

References

1. Tsai, W.C.; Hong, C.M.; Tu, C.S.; Lin, W.M.; Chen, C.H. A review of modern wind power generation forecasting technologies. *Sustainability* **2023**, *15*, 10757. [\[CrossRef\]](#)
2. Feng, C.; Cui, M.; Hodge, B.M.; Zhang, J. A data-driven multi-model methodology with deep feature selection for short-term wind forecasting. *Appl. Energy* **2017**, *190*, 1245–1257. [\[CrossRef\]](#)
3. Sun, H. Hybrid model with secondary decomposition, random forest algorithm, clustering analysis and long short memory network principal computing for short-term wind power forecasting on multiple scales. *Energy* **2021**, *221*, 119848. [\[CrossRef\]](#)
4. Sheoran, S.; Pasari, S. Efficacy and application of the window-sliding ARIMA for daily and weekly wind speed forecasting. *J. Renew. Sustain. Energy* **2022**, *14*, 053305. [\[CrossRef\]](#)
5. Aly, H.H.H. A hybrid optimized model of adaptive neuro-fuzzy inference system, recurrent Kalman filter and neuro-wavelet for wind power forecasting driven by DFIG. *Energy* **2022**, *239*, 122367. [\[CrossRef\]](#)
6. Hu, S.; Xiang, Y.; Huo, D.; Jawad, S.; Liu, J. An improved deep belief network-based hybrid forecasting method for wind power. *Energy* **2021**, *224*, 120185. [\[CrossRef\]](#)
7. Liu, Z.; Hara, R.; Kita, H. Hybrid forecasting system based on data area division and deep learning neural network for short-term wind speed forecasting. *Energy Convers. Manag.* **2021**, *238*, 114136. [\[CrossRef\]](#)
8. Wu, Z.; Zeng, S.; Jiang, R.; Zhang, H.; Yang, Z. Explainable temporal dependence in multi-step wind power forecast via decomposition based chain echo state networks. *Energy* **2023**, *270*, 126906. [\[CrossRef\]](#)
9. Hou, G.; Li, Q.; Huang, C. Spatiotemporal forecasting using multi-graph neural network assisted dual domain transformer for wind power. *Energy Convers. Manag.* **2025**, *325*, 119393. [\[CrossRef\]](#)
10. Chen, Z.; Zhang, B.; Du, C.; Meng, W.; Meng, A. A novel dynamic spatio-temporal graph convolutional network for wind speed interval prediction. *Energy* **2024**, *294*, 130930. [\[CrossRef\]](#)
11. Lv, Y.; Hu, Q.; Xu, H.; Lin, H.; Wu, Y. An ultra-short-term wind power prediction method based on spatial-temporal attention graph convolutional model. *Energy* **2024**, *293*, 130751.
12. Wu, X.; Zhan, H.; Hu, J.; Wang, Y. Non-stationary GNNCrossformer: Transformer with graph information for non-stationary multivariate spatio-temporal wind power data forecasting. *Appl. Energy* **2025**, *377*, 124492. [\[CrossRef\]](#)
13. Zhou, J.; Lu, X.; Xiao, Y.; Su, J.; Lyu, J.; Ma, Y.; Dou, D. SDWPF: A dataset for spatial dynamic wind power forecasting challenge at KDD Cup 2022. *arXiv* **2022**, arXiv:2208.04360. [\[CrossRef\]](#)
14. 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. [\[CrossRef\]](#)
15. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 22419–22430.
16. Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; Jin, R. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; pp. 27268–27286.

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.