IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 September 2023, accepted 18 November 2023, date of publication 21 November 2023,
date of current version 29 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3335629

RESEARCH ARTICLE

A Spatiotemporal Coupling Calculation-Based Short-Term Wind Farm Cluster Power Prediction Method

HAOCHEN LI[1,2], LIQUN LIU[1], AND QIUSHENG HE[1]
[1]School of Electronic Information Engineering, Taiyuan University of Science and Technology, Taiyuan 030024, China
[2]Department of Electrical and Control Engineering, Shanxi Institute of Technology, Yangquan 045000, China

Corresponding author: Liqun Liu (Liulq_1976@163.com)

This work was supported by the Shanxi Province Basic Research Program under Grant 202203021221153.

ABSTRACT Accurate short-term wind power prediction is of great significance to the real-time dispatching of power systems and the development of wind power generation plans. However, existing methods for wind power prediction have the following problems: 1) some studies aim at predicting wind power for a single wind farm, ignoring the correlations of the adjacent wind farms; 2) many studies tend to convert wind speed forecast results to wind power, increasing the conversion error; 3) almost all studies place emphasis on the capture of spatiotemporal features, neglecting the influence of spatiotemporal coupling. Therefore, to solve the above questions, this work proposes an adaptive graph neural network based on spatiotemporal attention calculation for short-term wind farm cluster power prediction, using only wind power data. Firstly, a dynamic undirected graph is established to sufficiently learn prior knowledge of spatial relationships. Next, the spatiotemporal coupling relationship and global temporal correlation between data can be computed by performing spatiotemporal cross-attention and temporal self-attention, respectively. Finally, a novel hybrid loss function is proposed to optimize the prediction model accurately. In a case study, compared with other benchmark methods, the proposed method shows excellent overall performance in predicting wind power.

INDEX TERMS Short-term wind power prediction, spatiotemporal coupling, adaptive graph neural network, attention calculation.

I. INTRODUCTION

With the goal of achieving carbon emission reduction by 2050, it has become the development trend of the current power supply to gradually replace traditional fossil fuel power generation with renewable energy [1]. Among all the renewable energy sources, wind energy is considered to be the most promising green source due to its low environmental pollution and large power generation capacity. According to the relevant data from the Global Wind Energy Council (GWEC), starting from 2023, the average annual installed capacity of wind power is expected to reach 136 GW in the next five years, and the new grid-connected capacity is expected to reach 680 GW [2]. However, the intermittence and stochasticity of wind power, as well as its high penetration of the power grid will inevitably affect the scheduling decisions of the grid and the development of wind farm generation plans [3]. To cope with the aforementioned adverse effects, developing a prediction algorithm that can accurately forecast wind power generation is an essential and challenging task.

Wind power prediction is typically divided into four types according to the time scale. These include ultra-short-term prediction (within 30 minutes), short-term prediction (30 minutes to 6 hours), medium-term prediction (6 hours to 1 day), and long-term prediction (1 day to 7 days) [4], respectively. Each type of prediction serves a different purpose, with the ultra-short-term prediction aiming at the real-time scheduling of the power system, the short-term prediction affecting both scheduling and generation planning, and the

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed A. Zaki Diab.

© 2023 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

medium and long-term predictions mainly applied to make the maintenance plan of wind turbines. In the practical operation of the power system, accurate short-term prediction is the focus of researchers.

After the development years, the mainstream prediction method can be classified into two categories: statistical models and intelligent models [5]. The former mainly analyzes historical data to obtain a mathematical model, which is used for making predictions. While the latter mainly refers to machine learning methods that rely on historical data to train models. These methods can learn the changing features of the data for prediction purposes. As a critical branch of machine learning, deep learning has shown outstanding performance in various territories, such as image processing, natural language processing, and others. Regarding prediction, deep learning also performs well. Compared to statistical models, generally speaking, intelligent models are better equipped to capture the nonlinear features of data variability, thereby achieving accurate predictions.

In recent years, the application of graph neural networks (GNN) has gained popularity due to their ability to map spatial features. Specifically, the recording of wind power-related data is carried out by the corresponding sensors, which can be represented as nodes, while the correlations (Euclidean distance, Pearson correlation, Granger causality, etc.) between these sensors are regarded as edges between nodes. In this manner, a graph with contained nodes and edges is established. It can be seen from the graphical construction process that the spatial correlations among the data can be better reflected by the graph. Moreover, the graph can be divided into a directed graph and an undirected graph based on the orientation of the edges, while it can also be divided into static and dynamic graphs based on whether the relationships between the nodes change or not. Apparently, from the construction process and categories of the graph, the prediction of wind power is not only related to the historical data of this node but also influenced by the data of neighboring.

Since wind generation is influenced by various meteorological factors (temperature, humidity, pressure, wind speed, etc.), many studies utilize these meteorological data to indirectly predict wind power. However, in terms of prediction of wind farm clusters, it is relatively difficult to collect multiple types of meteorological data, and with the increase in the amount of data, the consumption of computer resources becomes more pronounced. Therefore, in the case of using only wind power data, this paper draws inspiration from sparse spatial-temporal attention mechanism [6] and dynamic graph construction, proposing an adaptive graph neural network based on spatiotemporal attention calculation (AGSTA). The principal contributions of this paper are summarized as follows:

- The sparse spatial-temporal attention mechanism is simplified and an adaptive undirected graph is used as its calculation basis to accurately predict future data. At the same time, the addition of message passing facilitates the calculation of spatiotemporal coupling by spatiotemporal cross-attention, and is also more suitable for the processing of dynamic graphs.
- The model proposes a new hybrid loss function for multiple sites and multiple timesteps prediction. Under this model architecture, the loss function has certain advantages over the traditional loss function in terms of improving the overall prediction accuracy within 6h for a wind farm cluster.
- The model exhibits superior performance compared to other competing algorithms on the selected dataset. Specifically, the overall performance of the model outperforms the rest of the algorithms for wind farm cluster power prediction and remains the overall leader for individual wind farm power prediction.

The remaining sections of this paper are as follows: Section II primarily reviews recent research related to wind power prediction. Section III analyzes the impact of spatiotemporal coupling on the data. Section IV provides a detailed description of the proposed algorithm. Section V explains the model's hyperparameter selection and data processing steps, while Section VI focuses on experimental validation. Finally, Section VII provides a summary of the entire paper.

II. RELATED WORKS

In this section, the state-of-the-art algorithms in the field of prediction are summarized, with emphasis on the overview of GNNs.

A. STATISTICAL MODEL

Statistical models are essentially mathematical models, which are derived from the analysis of extensive historical data. The autoregressive model (AR) [7], autoregressive moving average model (ARMA) [8], and autoregressive integrated moving average model (ARIMA) [9] are the main representatives of such methods. Although the effectiveness of these methods for wind speed forecasting has been proved by a large number of studies, the drawbacks still exist. Specifically, these mathematical models are linear structure, which means they can only forecast the linear relationships in data, and cannot effectively capture the nonlinear features [10].

B. INTELLIGENT MODEL

Machine learning has proven to be highly effective in the prediction field due to its exceptional learning capability. When it comes to wind speed forecasting, machine learning can extract the relationships between historical data and then build a model to predict wind speed data in the future. At present, support vector machine (SVM) [11] is widely used to search the relationships between data by optimizing the structure of data. Building upon this, reduced SVM and least square SVM (ls-SVM) have been proposed by [12] and [13], respectively, which optimized the performance of SVM. Nevertheless, these approaches still possess certain limitations, such as reduced learning efficiency as dataset sizes increase.

Deep learning has been favored by researchers due to the rapid development of graphics processing units (GPUs) in recent years. Rather than superficial learning, it excavates deeper information from the data. In the field of prediction, [14] and [15] drew inspiration from restricted boltzmann machines (RBM) and rough set theory, leading to a redesign of neuron structures that enable the exploration of deeper temporal information in data. Additionally, long-short time memory (LSTM) [16] and gated recurrent units (GRUs) [17] are also important methods for time series prediction. These methods facilitate the memory and transmission of information through the gated unit, enabling accurate prediction. While these methods above excel at capturing the local temporal information of the data, they cannot do anything about the spatial information of the sequence.

Consequently, [18] and [19] proposed hybrid models based on LSTM. The former combined dictionary learning (DL) with LSTM and integrated it into a sequence-to-sequence (seq2seq) architecture, enhancing the model's ability to extract temporal features while also improving spatial feature extraction. The latter combined CNN and LSTM, leveraging the strengths of both. However, the performance of the former in prediction tasks requires further validation, and the latter necessitates the collaboration of multiple types of data for accurate prediction.

C. GRAPH NEURAL NETWORK

References [20] and [21] introduced and developed the concept of graph networks, respectively. A GNN represents the data in a graphical form and processes them using a graphical approach. At the same time, this method has good interpretability in comparison to the conventional deep learning methods. With the continuous improvements in GNN, it has become increasingly popular in prediction. Nowadays, the research on GNN mainly concentrates on the following two aspects based on the extraction of spatiotemporal features.

On one hand, in order to extend convolution to graph structures, [22] proposed graph convolutional networks (GCN) from both spatial and spectral perspectives. However, both of these methods are relatively complex in structure and slow in operation. Another convolutional algorithm is diffusion convolution (DC) [23]. This method combines DC with GRU in the seq2seq architecture to enhance the extraction of spatiotemporal information. Nevertheless, this approach is computationally expensive and relatively resource-intensive. Graph wavenet (Graph WN) [24] and multivariate time series graph neural networks (MTGNN) [25] addressed adaptive graph structures by leveraging combinations of stacked 1D convolution components and improved graph convolution, including enhanced temporal convolution, to mine spatiotemporal relationships within the data. In addition, [26] constructed a directed graph based on granger causality and embed multiscale temporal convolutions into improved graph convolutions to capture spatiotemporal features among wind farms. However, the fundamental components of these models proposed in the three methods mentioned above are all CNN, which are not suitable for extracting the relevant characteristics of data in non-Euclidean space. Reference [27] constructed a spatiotemporal correlation matrix, which can be used to explore dynamic correlations between neighboring wind farm data. Yet, this method ultimately still extracts the spatiotemporal characteristics of the data in the form of GCN, which limits the model's ability to extract global features.

On the other hand, an increasing number of studies tend to introduce attention mechanisms into GNN calculation. Within the realm of prediction, the attention mechanism is able to compute the similarity between data and thus grasp the pattern of data change from a global perspective. In one study [4] combined GCN with maximum information and merged the self-attention mechanism with multiscale convolution kernels to enhance the ability to extract spatiotemporal information. Another study [28] developed a novel graph utilizing temporal polynomials. On this basis, the attention mechanism was coupled with seq2seq architecture to realize multivariate time series prediction. In [29], it employed spectral domain convolution and channel-wise attention to uncover the potential spatiotemporal dependencies to achieve multi-node wind speed forecasting. Reference [30] proposed an approach for traffic flow forecasting. This method merged spatial convolution with the learnable positional attention mechanism [31] to efficiently aggregate information from neighboring nodes, enabling the capture of spatiotemporal features between traffic flow data. Furthermore, [32] designed a dual-path architecture to capture both existing and potential spatial characteristics in traffic flow, and utilized a graph attention mechanism [33] to make the model applied to inductive learning tasks. However, these aforementioned model structures are complex, involve a large number of adjustable parameters, and consume significant computational resources.

Table 1 provides a summary of GNN-related methods mentioned above. It is evident that the majority of these methods construct their models using either GCN or CNN, both of which have noticeable shortcomings. The proposed AGSTA addresses the coupling relationships in the data, which have been overlooked in prior researches, through the spatiotemporal attention calculations, and utilizes the message passing mechanism to theoretically simulate the effects of convolutions, which can effectively avoid the inherent shortcomings of GCN and CNN.

III. DYNAMIC SPATIOTEMPORAL CORRELATION

In this section, the coupling relations of wind power data between neighboring wind farms are mathematically demonstrated to emphasize the necessity of constructing the

TABLE 1. Summary of different GNN methods.

Methods	Model basis	Graph structure	Advantage	Potential issues
Ref. [23]	DC+GRU	Static undirected graph	Capturing distant relationships Enhancing model robustness	Computationally complex
Ref. [24]	CNN+DC	Dynamic self-adaptive graph	Stronger representational capacity	Limited in non-Euclidean space
Ref. [25]	CNN			
Ref. [26]		Dynamic directed graph	Clear causal relationships Stronger representational capacity	
Ref. [27]	GCN	Hierarchical directed graph	Relatively lower computational complexity Strong ability to capture local information	Limited ability to capture global information
Ref. [28]	CNN+ Attention	Temporal polynomial graph	Multi-scale feature extraction Capturing cross-channel relationships	Complex model Strong data dependency Large computational overhead Difficult parameter tuning
Ref. [4]	GCN+CNN +Attention	Static undirected graph	Multi-scale feature extraction Integration of cross-modal information	
Ref. [30]	GCN+GRU +Attention	Dynamic undirected graph	Multi-scale feature extraction Effective spatiotemporal modeling	
Ref. [29]	Spectral GCN +Attention	Static undirected graph	Detailed relationship modeling Better feature propagation	
Ref. [32]	CNN+Graph Attention	Dynamic + Static	Multi-scale feature extraction Fusion of global and local information	

dynamical adjacency matrix and a detailed procedure for constructing is presented.

A. DATA STRUCTURE ANALYSIS

The data in this paper is taken from the Wind Integration National Dataset. Six adjacent stations are selected from Colorado and New Mexico, each of them is a wind farm covering an area of 2KM × 2KM, and the distance between stations is calculated using the longitude and latitude of each station. The detailed geographical locations are shown in Figure 1.

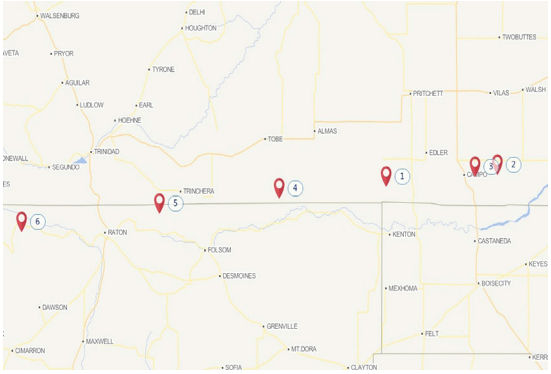


FIGURE 1. Geographical coordinate distribution map of six stations.

The nodes in Figure 1 are wind farms, and the input data in each node is the historical wind power data for that node.

B. SPATIOTEMPORAL COUPLING ANALYSIS

The spatiotemporal correlation exists among the meteorological conditions of adjacent wind farms [26]. The influence of meteorological conditions establishes a correlation between the wind power data, giving rise to spatiotemporal coupling. The primary manifestation of this coupling is the establishment of dynamic relationships among the wind power data. To quantitatively analyze this relationship, the paper employs Normalized Mutual Information (NMI).

NMI is used to judge the similarity between the algorithm and standard results, that is, the amount of information contained in one set of data about the other.

The NMI calculation can be divided into two steps: the mutual relationship calculation and the normalization process.

The first step is to compute the mutual relationships between variables, the formulation is shown as (1).

$$\mathrm{MI}(X, Y) = \sum_{i=1}^{N}\sum_{j=1}^{M} P(X_i, Y_j)\log(\frac{P(X_i, Y_j)}{P(X_i)P(Y_j)}) \tag{1}$$

where X and Y represent two sets of data that need to be evaluated for similarity, N and M are the number of variables X and Y, i and j represent ith and jth variable, $P(X_i)$ and $P(Y_j)$ are the probabilities of X_i and Y_j, $P(X_i, Y_j)$ denotes the joint probabilities of X_i and Y_j, and MI is the computation result of mutual information.

The second step is to normalize mutual information, the expression is (2).

$$\mathrm{NMI}(X, Y) = \frac{2\mathrm{MI}(X, Y)}{H(X) + H(Y)} \tag{2}$$

where $H(X)$ and $H(Y)$ are the information entropy of X and Y, which are presented in the form of (3).

$$H(X) = \sum_{i=1}^{N} P(X_i)\log\frac{1}{P(X_i)} \tag{3}$$

The value of NMI is [0,1], which means if the value is closer to 1, the higher the similarity between X and Y, and vice versa, the smaller the similarity between them.

The results of NMI calculations are shown in Figure 2. The Figure shows the similarity of wind power between the six stations.

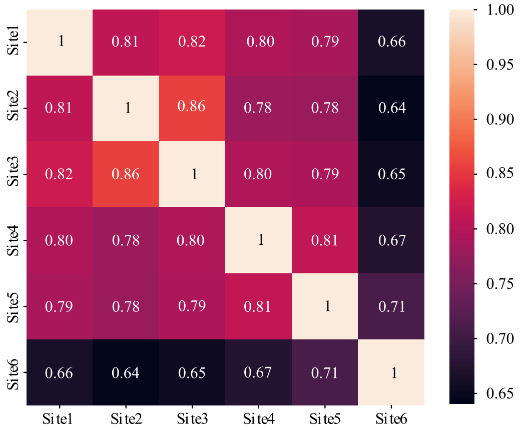


FIGURE 2. **Wind power data similarity heatmap.**

Figure 2 shows that the similarities between site 1 and sites 2, 3, and 4 are strong. This is further supported by Figure 1, which indicates that site 1 is located in the center of these three sites. The similarity between sites 2 and 3 is the strongest among all sites, which is due to the fact that the two sites are closest to each other in distance, this proximity leads to the meteorological factors being less variable over short distances. Similarly, sites 4 and 5, along with sites 5 and 6, reveal the strongest similarity compared to other sites. It can be inferred that the closer the stations are, the higher the similarity between them, and vice versa, the lower the similarity. For example, sites 2 and 6, which are the farthest apart among all sites, have the lowest NMI value of 0.64.

Based on the traditional adjacency matrix construction method in GNN, the sites can be viewed as nodes, the sites with high similarity are connected by edges, and the distance between nodes is considered as the weight of edges. With this approach, the static graph can be constructed to perform GNN computation. However, it is further observed from Figure 2 that the inter-site NMI values are not equal to 1, even for the most correlated sites, except for themselves. This phenomenon indicates that there is different information in wind power data from each other. In other words, there are similarities, but also differences, in the variability patterns of wind power.

From the above, it is clear that the conventional way of constructing an adjacency matrix does not adequately express the correlations between nodes, because these similarities are time-varying rather than static. As a solution, this work constructs an adaptive adjacency matrix instead of a static one.

C. ADAPTIVE ADJACENCY MATRIX

The dynamic adjacency matrix is constructed based on the static adjacency matrix. According to the distance between stations, a weight matrix can be constructed, which can then be normalized by the Gaussian kernel function, and the normalized result is shown in (4).

$$\mathbf{W} = \begin{cases} \text{weights}, & \text{weights} \geq \text{threshold} \\ 0, & \text{weights} < \text{threshold} \end{cases} \tag{4}$$

where weights represent the normalized distance between the nodes, threshold with a value of 0.01, and $\mathbf{W}$ is the weight matrix. By setting a threshold for the normalized distance, correlations that fall below this value can be safely ignored, simplifying the following attention calculations and reducing unnecessary computer resources.

For the construction of a dynamic graph, this paper refers to papers [23] and [24]. The $\mathbf{W}$ is subjected to Laplace transform, it can be depicted by (5).

$$\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{I} + \mathbf{W})\mathbf{D}^{-\frac{1}{2}} \tag{5}$$

where $\mathbf{I}$ denotes the identity matrix, $\mathbf{D}$ is the degree matrix, and $\mathbf{L}$ is the Laplace matrix. In addition, the DC can be generalized to the undirected graph, resulting in (6).

$$\boldsymbol{Z} = \sum_{k=0}^{K-1} \mathrm{P}^{k}\mathrm{X}\boldsymbol{W}_{k1} \tag{6}$$

where X denotes the input signal, $\boldsymbol{W}$ is the learnable parameter matrix, k is the diffuse steps, K is the sum number of diffusion steps, $\mathbf{P}$ is the transfer matrix of the diffusion process, and $\boldsymbol{Z}$ is the diffuse convolution result.

A singular value decomposition is implemented for $\mathbf{P}$, as shown in equation (7).

$$\mathbf{P} = \mathbf{U}\mathrm{diag}(\mathbf{S})\mathbf{V}^{\mathrm{T}} \tag{7}$$

where $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{S}$ are the matrices from the singular decomposition.

Finally, the adaptive adjacency matrix is shown as (8).

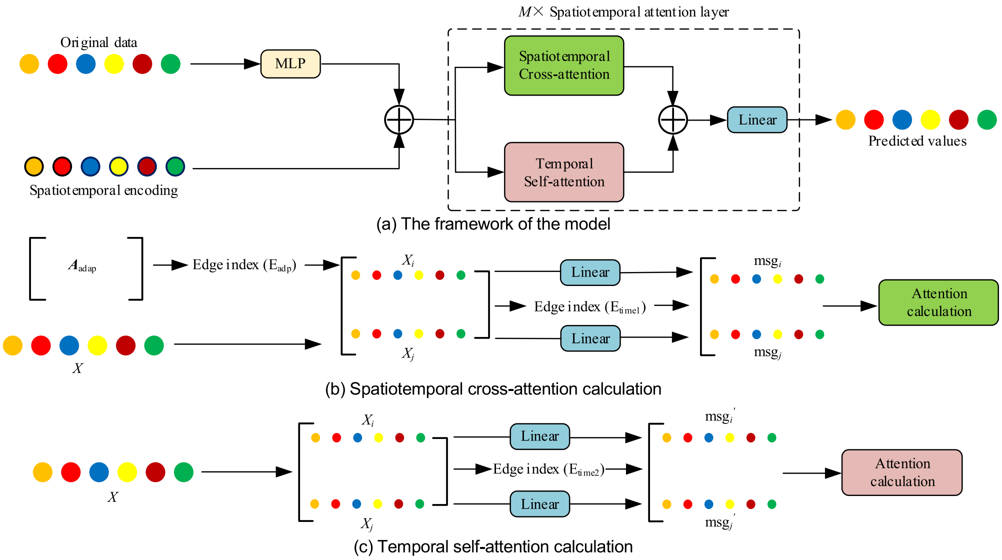
$$\mathbf{A}_{\mathrm{adap}} = \mathrm{sofmax}(\mathrm{ReLU}(\boldsymbol{E}_1\boldsymbol{E}_2^{\mathrm{T}})) + \mathbf{L} \tag{8}$$

where $\boldsymbol{E}_1$, and $\boldsymbol{E}_2$ are the learnable parameter matrices, $\boldsymbol{E}_1 = \boldsymbol{U} \times \sqrt{\mathrm{diag}(\mathbf{S})}$, $\boldsymbol{E}_2 = \sqrt{\mathrm{diag}(\mathbf{S})} \times \boldsymbol{V}^{\mathrm{T}}$.

The adaptive adjacent matrix is a matrix that can be updated during the network training process. It can better reflect the time-varying similarities between nodes and lay the foundation for the subsequent computation.

IV. METHODOLOGY

In this section, the AGST model is elaborated, including the principle of spatiotemporal attention calculation, and the optimization mechanism of the hybrid loss function.



(a) The framework of the model

(b) Spatiotemporal cross-attention calculation

(c) Temporal self-attention calculation

FIGURE 3. The framework of spatiotemporal attention calculation.

A. MODEL OVERVIEW

Existing GNN algorithms focus their research on the extraction of spatiotemporal features, either by improving spatial convolution or using attention calculation. However, they often overlook a vital characteristic, namely, the coupling relationship between time and space in wind power data. To achieve better prediction results, this study replaces the skip connection and MLP in the original algorithm with Linear layers, simplifying the structure of sparse spatiotemporal attention, and introduces edge weight aggregation. The overall model framework is shown in Figure 3 (a).

In Figure 3 (a), MLP is a multi-layer perception, which consists of the linear layer and the dropout. Before the raw data is fed into the network operation, the data needs to be spatiotemporally encoded in a separate network that can be learned during the training process. Then, the encoded data is sent into the spatiotemporal attention network, which is divided into two parts. The first part is spatiotemporal cross-attention, which is used to compute the coupling relationships as a way to identify the specific correlations of the data in time and space. The other part is the temporal self-attention, which is used to capture the global relationships of the data at the temporal level. Ultimately, the spatiotemporal network produces its final prediction by combining the results from two attention blocks. It is worth noting that the spatiotemporal network can be stacked, and multiple attention calculations can be carried out by the stacked layers, enabling multiple propagations and aggregations of messages and eventually satisfactory results.

B. SPATIOTEMPORAL ENCODING

AGSTA differs from recurrent neural networks (RNN) in that it does not inherently capture temporal features in data, thus necessitating the incorporation of temporal encoding. To construct a learnable temporal encoding, the absolute positional encoding that comes from [31] is adopted here. As is shown in (10).

$$E_{\mathrm{t}} = \begin{cases} \sin(t/10000^{2i/d_{\mathrm{model}}}) \\ \cos(t/10000^{(2i+1)/d_{\mathrm{model}}}) \end{cases} \tag{9}$$

where t is the time step, d_{model} is the dimension of the encoding, $2i$ and $2i+1$ denote even and odd dimensions, and E_{t} is the encoding used for learning.

Furthermore, AGSTA requires encoding of not only time but also the spatial locations of six stations. To this end, this work utilizes random initialized vectors to encode spatial position. Ultimately, spatiotemporal encoding is the sum of temporal and spatial encoding, as specified in (10).

$$\mathrm{Enc}_{\mathrm{st}} = E_{\mathrm{t}} + E_{\mathrm{s}} \tag{10}$$

where E_{s} is the spatial encoding and $\mathrm{Enc}_{\mathrm{st}}$ is the final spatiotemporal encoding.

By setting the spatiotemporal coding, the network is able to more accurately identify the relative position and temporal order between data.

C. SPATIOTEMPORAL CROSS-ATTENTION

In Figure 3 (b), X is the data with encoding, X_i and X_j are the target and source node information divided by the edge index from the adaptive adjacency matrix, where the edge index represents the spatial dimension. Similarly, msg_i and msg_j denote the target and source node information indexed by the edge from the temporal dimension. Notably, cross-attention means the attention calculation of data between the spatial and temporal dimensions. The detailed procedure is elaborated below.

After the data X and $\boldsymbol{A}_{\mathrm{adp}}$ enter the network, the indices of edges in $\boldsymbol{A}_{\mathrm{adp}}$, denoted as $\mathrm{E}_{\mathrm{adp}}$, are used as a reference to segment X_i and X_j. Then, the new edge indices $\mathrm{E}_{\mathrm{time1}}$ are constructed in the time dimension of X_i and X_j. Using $\mathrm{E}_{\mathrm{time1}}$

as a reference, a linear transformation is applied to X_i and X_j, and the resulting values are divided to obtain the source node and the target node information, msg$_j$ and msg$_i$, for message passing. From the whole procedure, it is not difficult to see that msg$_i$ and msg$_j$ contain both spatial and temporal information. Finally, the attention calculation is implemented by msg$_i$ and msg$_j$. The attention calculation here is inspired by [6] and [33], and the specific calculations are presented in equations 11, 12, and 13, respectively.

$$\mathrm{msg}_{\mathrm{sum}} = \mathrm{MLP}(\sum \mathrm{msg}_{t\to r}^{j\to i}) \tag{11}$$

where t and r denote the time steps, and the symbol → denotes the message aggregation direction or time flow direction. The equation represents that the message is propagated from node j to node i over the time interval from t to r. At node i, the messages are summed and then passed through an MLP to form the final aggregated message.

$$\alpha = \frac{\exp(\mathrm{msg}_{\mathrm{sum}} \cdot \boldsymbol{w})}{\sum\limits_{\mathrm{msg} \in \mathrm{msg}_s^{j\to i}} \exp(\mathrm{msg} \cdot \boldsymbol{w})} \tag{12}$$

where s denotes the time step between [t, r], $\boldsymbol{w}$ is a learnable weight matrix, α is the information score after the softmax layer, and msg is the result obtained by performing a maximum operation on msg$_{\mathrm{sum}}$ based on the indices of target node edges in E$_{\mathrm{time1}}$, similar to pooling.

$$e = \alpha \cdot \mathrm{msg}_{\mathrm{sum}} \tag{13}$$

where e denotes the context vector, which is the result of the final attention calculation, and reflects the similarity between the information of node i and node j.

After the attention computation, two rounds of information propagation are required. First, e needs to be propagated and updated along the edges in E$_{\mathrm{time1}}$, so that the results of the attention computation can be diffused to target nodes. After that, a second propagation is needed, where the weights of edges in E$_{\mathrm{adp}}$ are aggregated towards the target nodes. The second aggregation is shown as (14).

$$\mathrm{ST}_2 = \mathrm{ST}_1 \times \mathrm{weights} \tag{14}$$

where weights refer to the weights of edges in $\boldsymbol{A}_{\mathrm{adp}}$. ST$_1$ represents the result after the first aggregation update, while ST$_2$ denotes the value that requires executing the second propagation.

D. TEMPORAL SELF-ATTENTION

In comparison to the spatiotemporal cross-attention calculation, the temporal self-attention calculation is relatively simple. It only requires the edge indices E$_{\mathrm{time2}}$ as the reference for partitioning msg$_i$' and msg$_j$' for the attention computation. The expression of the self-attention calculation is the same as that of the spatiotemporal cross-attention, as is shown in (11), (12) and (13). In addition, the information in this process only needs to be propagated once after the end of the attention calculation to obtain the final output, Temp.

E. HYBRID LOSS FUNCTION

The traditional loss function is constructed based on the mean absolute error (MAE) [23], [24], [25], [26], [30]. This loss function can optimize the performance of the model relatively comprehensively, so as to obtain satisfactory results, as shown in equation (15).

$$\mathrm{MAE} = \frac{\sum\limits_{i=1}^{N} \left| p_{\mathrm{predicted}i} - p_{\mathrm{true}i} \right|}{N} \tag{15}$$

where $p_{\mathrm{predicted}}$ denotes the predicted values from the model, p_{true} is the true value, i denotes the ith predicted value or the ith true value, and N is the total amount of data.

Taking into consideration of the dataset used in this paper is composed of only one type of wind power data, it is more singular compared to the types of data used in other studies. In this case, inspired by [27], a different loss function is proposed for AGSTA, which is stipulated in equations (16), (17), and (18). Under the influence of this function, the model is expected to yield more accurate prediction results.

$$L_{\mathrm{time}} = \frac{1}{T} \sum_{i=1}^{T} \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left| p_{\mathrm{prediceted}i} - p_{\mathrm{true}i} \right|} \tag{16}$$

where N denotes the number of stations, T denotes the predicted timesteps, i denotes the ith station or the ith timestep, and L_{time} denotes the temporal loss function.

$$L_{\mathrm{space}} = \frac{1}{N} \sum_{i=1}^{N} \sqrt{\frac{1}{T} \sum_{i=1}^{T} \left| p_{\mathrm{predicted}i} - p_{\mathrm{true}i} \right|} \tag{17}$$

where L_{space} represents the spatial loss function.

$$L = \alpha \cdot L_{\mathrm{time}} + \beta \cdot L_{\mathrm{space}} \tag{18}$$

where α and β represent penalty coefficients of L_{time} and L_{space}, respectively, and the values of α and β are between [0,1]. The advantage of this loss function is that it can fully take into account multi-site prediction situations, and can predict more accurate results.

F. ERROR EVALUATION METRICS

In this paper, three error evaluation metrics are chosen to measure the prediction effectiveness, which are MAE, root mean square error (RMSE), and weighted average percentage error (WAPE). The formula for MAE is shown in (16), and it reflects the average difference between the predicted and true values by offsetting the positive and negative errors against each other in absolute terms. the equations for RMSE and WAPE are shown in (19) and (20).

$$\mathrm{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (p_{\mathrm{predicted}i} - p_{\mathrm{true}i})^2} \tag{19}$$

where RMSE tends to represent stability, as it reflects whether the predicted curve can maintain a relatively slight error to the

Algorithm 1 The Training Procedure of AGSTA

```
Input: The training data $P_{training}$, and the distance matrix $\mathbf{W}$
Output: Trained model AGSTA
For each training epoch do:
    Calculate the adaptive adjacency matrix $\boldsymbol{A}_{adp}$ according to (5-8)
    Calculate the spatiotemporal encoding $Enc_{st}$ according to (9-10)
    $X \leftarrow P_{training} + Enc_{st}$
    For $i = 0$ to $M$ do:
        // Compute 1 and 2 in parallel
        1. The spatiotemporal cross-attention calculation
         Input: $X$, edge index $E_{adp}$ and edge weights of $\boldsymbol{A}_{adp}$
        Output: The result of cross-attention calculation $ST_2$
        Based on $E_{adp}$, obtain source node $X_j$ and target node $X_i$
        Based on $E_{time1}$, obtain source node $msg_j$ and target node $msg_i$
        Perform cross-attention calculation according to (11-13)
        Perform two rounds of message propagation along $E_{time1}$ and
      $E_{adp}$ respectively to obtain $ST_2$
        2. The temporal self-attention calculation
        Input: $X$
        Output: The result of self-attention calculation, Temp
        Based on $E_{time2}$, obtain source node $msg_j$' and target node $msg_i$'
        Perform self-attention calculation according to (11-13)
        Perform one message propagation along $E_{time2}$ to obtain Temp
        3. The combination of $ST_2$ and Temp
        $X \leftarrow$ Linear ($ST_2$ + Temp)
    End for
    Based on (16-18), construct the loss function $L$ using the output of the $M$th layer
End for
```

true curve at each time instant without large fluctuations.

$$\text{WAPE}=\frac{\sum_{i=1}^{N}\left|p_{\text{true}i}-p_{\text{predicted}i}\right|}{\sum_{i=1}^{N}\left|p_{\text{true}i}\right|}\times 100\% \tag{20}$$

where WAPE reflects the proportion of the total error in the true value.

Until now, the AGSTA model has been constructed successfully. Meanwhile, the overall training procedure is summarized in Algorithm 1.

V. MODEL DETAILS

This section elaborates on the selection method for important parameters within the model, as well as the model's data processing process.

A. ABLATION EXPERIMENT

This study employs the method of controlling variables and utilizes MAE and RMSE as the primary indicators to determine the significant parameters of the model.

Figure 4 illustrates the impact of different batch sizes on model performance. As evident from the graph, MAE is relatively unaffected by changes in batch size, whereas RMSE increases with larger batch size. When the batch size is 8, RMSE is minimized, indicating the highest overall fit between the predicted and actual curves. Generally, larger batch sizes lead to reduced model accuracy, while smaller batch sizes increase computational time. Taking these factors into account, this paper selects a batch size value of 128.

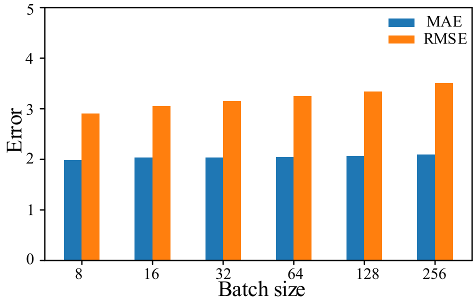


FIGURE 4. MAE and RMSE with different batch sizes.

Figure 5 depicts the relationship between the number of spatiotemporal (ST) layers and model performance. As shown in the graph, with an increase in the number of ST layers, the overall trend of the error gradually decreases. This is due to the repetitive entry of data into the ST layers, enabling repeated calculations of their coupling and temporal relationships, thereby achieving more accurate predictions. However, an increase in the number of ST layers would lead to higher utilization of hardware resources by the model, impacting its computational efficiency. In this context, the number of ST layers is set to 2.

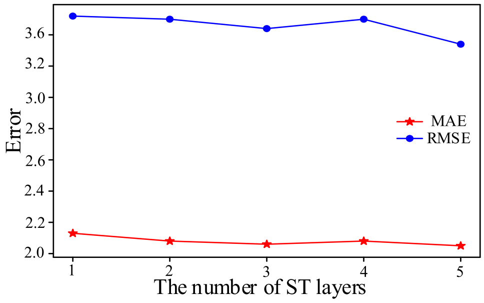
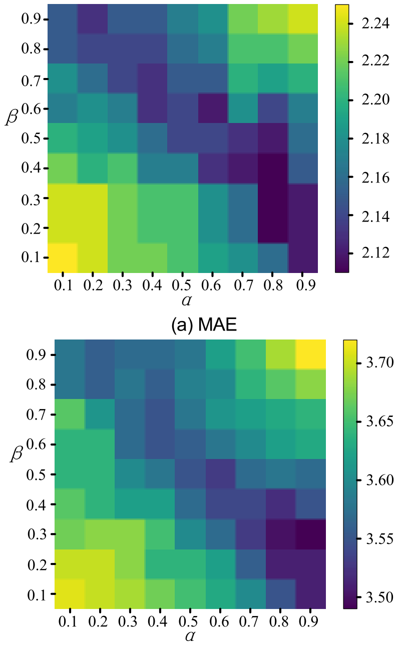


FIGURE 5. MAE and RMSE with different ST layers.

Figure 6 illustrates the impact of the loss function penalty coefficients, α and β, on model performance. It is evident that when $\alpha = 0.8$ and $\beta = 0.2$ to 0.5, the MAE metric is minimized, while when $\alpha = 0.9$ and $\beta = 0.3$, RMSE is minimized. This indicates that the model is more inclined towards optimizing for time. However, a comparison of errors corresponding to $\alpha = 0.1$, $\beta = 0.9$, and $\alpha = 0.1$, $\beta = 0.1$ reveals that optimizing for space can also enhance the model's predictive accuracy. Therefore, α and β are chosen as 0.9 and 0.1, respectively.



(a) MAE

(b) RMSE

FIGURE 6. **MAE and RMSE with different α and β.**

Figure 7 shows how the loss function decays with the increment of iteration, under the chosen value of α and β. The loss function decreases rapidly as the number of iterations increases. When the number of iterations is more than 50, its loss function tends to be a constant value, indicating that AGSTA has converged. During the whole process of training, the training process of AGSTA is relatively stable, and no gradient vanishing problem makes it difficult to converge.

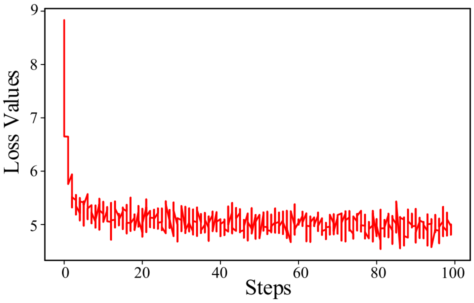


FIGURE 7. **Training process of AGSTA.**

Table 2 shows the impact of different optimizers on model performance. Based on the data, it is evident that Adam and RMSprop exhibit significantly better overall performance compared to other optimizers. Specifically, in terms of the MAE metric, the performance of both optimizations remains consistent. However, when considering the RMSE and WAPE metrics, the former significantly outperforms the latter. Therefore, Adam optimizer is chosen for this paper.

TABLE 2. **Performance comparison of different optimizers.**

Optimizer	MAE	RMSE	WAPE
Adadelta	4.93	6.66	87.81%
Adagrad	3.56	4.83	64.36%
SGD	4.69	6.43	83.56%
Adamax	2.18	3.58	39.83%
RMSprop	**2.13**	3.53	39.04%
Adam	**2.13**	**3.51**	**39.00%**

B. DATA PROCESSING PROCESS

The detailed data processing procedure of the model for the given data is shown in Figure 8. The initial input data has a shape of 31,267 × 6, where 6 refers to 6 sites, and 31,267 is the total time steps.

In the first step, the data is preprocessed. In this step, the sliding window model is primarily used to adjust the data to the required shape for the model, which is 128 × 12 × 6 × 1. Here, 128 corresponds to the batch size during data processing; 12 refers to the number of time steps, which can be adjusted based on the predicted time range; and 1 represents the feature dimension.

In the second step, the spatiotemporal encoding network is operated. The spatiotemporal encoding of the data consists of two components: the time encoding E_t and the spatial encoding E_s. They correspond to the time dimension and spatial dimension of the input data, respectively. Here, E_t is of size $12 \times 1 \times 32$, while E_s is of size 6×32. The resulting Enc_{st} is $128 \times 12 \times 6 \times 32$.

In the third step, spatiotemporal attention is computed. For cross-attention, the process starts by referencing $\mathrm{E}_{\mathrm{adp}}$, resulting in X_i and X_j with shapes of $128 \times 12 \times 36 \times 32$. Here, $\mathrm{E}_{\mathrm{adp}}$ has a shape of 2×36. Then, referencing $\mathrm{E}_{\mathrm{time1}}$, msg_i and msg_j are obtained with shapes of $128 \times 144 \times 36 \times 32$, where $\mathrm{E}_{\mathrm{time1}}$ has a shape of 2×144. Subsequently, attention calculation is performed following equations (11-13). Finally, two rounds of message propagation are conducted along $\mathrm{E}_{\mathrm{time1}}$ and $\mathrm{E}_{\mathrm{adp}}$ respectively, yielding the cross-attention result with a shape of $128 \times 12 \times 6 \times 32$, denoted as ST_2. As for self-attention, its operational mechanism is largely similar to cross-attention, with the distinction that it undergoes message propagation only once along $\mathrm{E}_{\mathrm{time2}}$, resulting in Temp having the same shape as ST_2.

In the final step, the predictions are obtained. By summing ST_2 and Temp, and processing them through a linear layer,

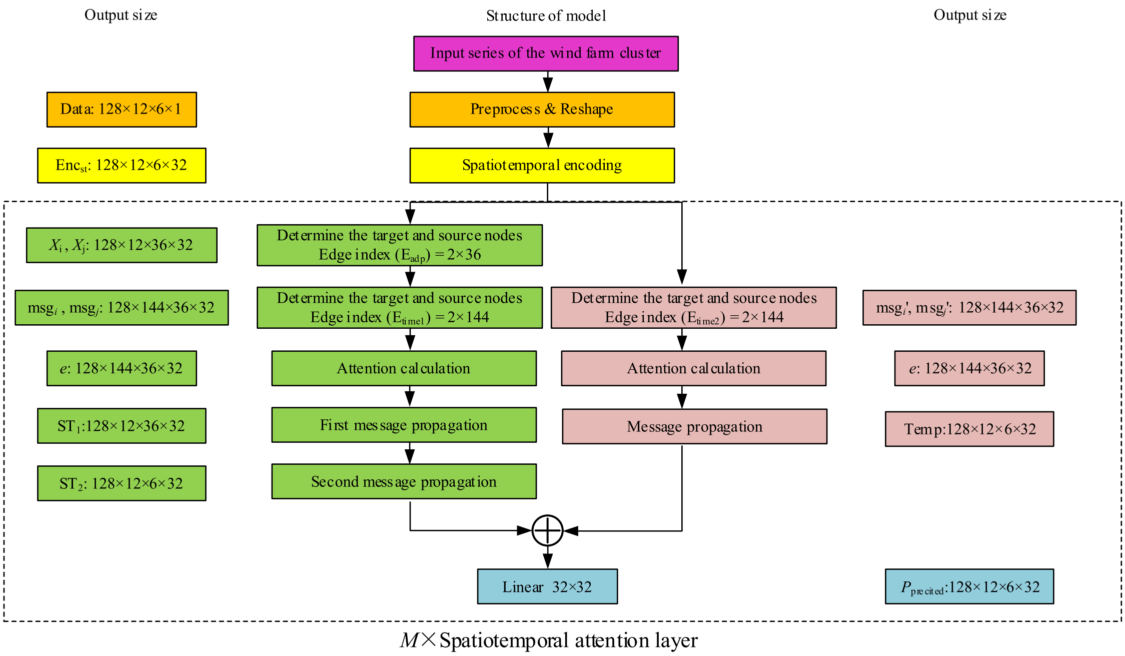

Output size
Structure of model
Output size
Input series of the wind farm cluster
Data: 128×12×6×1
Preprocess & Reshape
Enc$_{st}$: 128×12×6×32
Spatiotemporal encoding
X_i, X_j: 128×12×36×32
Determine the target and source nodes Edge index (E$_{adp}$) = 2×36
msg$_i$, msg$_j$: 128×144×36×32
Determine the target and source nodes Edge index (E$_{time1}$) = 2×144
Determine the target and source nodes Edge index (E$_{time2}$) = 2×144
msg$_i$', msg$_j$': 128×144×36×32
e: 128×144×36×32
Attention calculation
Attention calculation
e: 128×144×36×32
ST$_1$:128×12×36×32
First message propagation
Message propagation
Temp:128×12×6×32
ST$_2$: 128×12×6×32
Second message propagation
Linear 32×32
$P_{precited}$:128×12×6×32
M× Spatiotemporal attention layer

FIGURE 8. Data processing process.

the ultimate prediction values can be obtained. It's noteworthy that the shape of the final prediction values remains the same as the input shape into the network, which is 128 × 12 × 6 × 32. This design facilitates the repetitive execution of spatiotemporal attention computation.

VI. EXPERIMENT RESULTS

In this section, corresponding simulations are conducted to compare the performance of the state-of-the-art algorithms in the short-term prediction of wind power.

A. DATASET

According to the introduction of [35], the wind integration national dataset comprises meteorological data collected throughout the United States, including wind speed, wind direction, air density, air pressure, etc. Then, based on the actual geographical conditions of the United States near the sea and inland, 126,684 sites that can be used for the construction of wind farms are established, each site covers an area of 2km × 2km. Finally, by using these meteorological data, the wind power of wind turbines at different heights with different time resolutions is simulated and calculated at each site. In other words, the meteorological data are real-time data, while the power data are the results obtained from simulations. Under the above conditions, the land utilization rate is not the same for each site and the wind energy capacity factor is also different for each site. Hence, the most realistic effect can be simulated as much as possible.

In this study, the wind power data are obtained for wind turbines with hubs height of 100 m and the time resolution of 15 minutes, with a period is January 19, 2013, to December 31, 2013. 80% of the total data are set as training data for the proposed model, 10% are used to verify the validity of the model, and the remaining data are used as test data.

Table 3 presents the environmental conditions of the six selected sites in this study. It is evident that the average wind speed and capacity factor of WF4 and WF6 are significantly lower than the other sites. This will result in a considerable reduction in their wind power output, posing a challenge to the predictive performance of the model.

TABLE 3. The environmental condition of each site.

Environment	WF1	WF2	WF3	WF4	WF5	WF6
Usable area	1.00	1.00	1.00	1.00	0.90	0.96
Capacity (MW)	16	16	16	16	16	16
Wind speed (m/s)	7.64	8.33	8.25	6.87	9.12	6.24
Capacity factor	0.38	0.45	0.44	0.30	0.47	0.25

B. BASELINE

SVR [12]: The basic idea of SVR is to find the separated hyperplane that can correctly partition the training dataset and minimize the distance to the sample points farthest from that plane.

IPDL [15]: This model is built upon the RBM and rough set theory. The former learns the probability distribution of the dataset to capture the unsupervised temporal characteristics within wind speed data, while the latter enhances the model's robustness.

TABLE 4. Hyperparameter settings of models.

Models	Hyperparameter
IPDL	Epoch: 300; learning rate: 0.001; α: 0.8; β: 0.2; batch size: 64
RAE	Epoch: 200; learning rate: 0.001; α:0.8; β: 0.2; batch size: 64
Deep Forecast	LSTM layers: 2; epoch: 100; learning rate: 0.001; batch size: 128
DTDL	Sparse coefficient vector: 128×6×6; epoch: 100; learning rate: 0.001; batch size: 128
Graph WN	ST layers: 2; temporal convolution: (32, 32, (1, 2), 1); diffusion steps: K=2; epoch: 100; learning rate: 0.0001; batch size: 128
MTGNN	The kernel of dilated inception: [8, 9, 12, 13]; mix hop-propagation: (96, 32, (1, 1), (1, 1)); epoch: 100; learning rate: 0.0001
STGNN	Heads: 6; d_k=d_v=16; d_{model}=96; epoch: 100; learning rate: 0.0001; batch size: 128

RAE [16]: Similar to IPDL, the difference is that this model employs a seq2seq architecture to learn the unsupervised features within wind speed data.

Deep Forecast [17]: This model is essentially based on LSTM, achieving the extraction of spatiotemporal information from multiple wind farms' wind speed data through the stacking of multiple LSTM layers.

DTDL [19]: This model combines DL, LSTM, and AE. By utilizing LSTM-AE, it captures the deep temporal patterns within the electrical signals, and through DL, it identifies the most representative atoms within this pattern.

Graph WN [26]: In order to accurately capture the temporal relationships among various nodes, this model constructs an adaptive graph. Subsequently, it utilizes dilated causal convolution to build the gated temporal convolutional layer and combines it with diffusion convolutions to extract the spatiotemporal information within the traffic flow data.

MTGNN [27]: This model constructs mix-hop propagation and dilated inception layers through the stacking and combination of convolutional layers. Leveraging mix-hop propagation and dilated inception layers, it creates the graph convolutional module (GC) and temporal convolutional module (TC) respectively. By interleaving GC and TC and coupling them with graph learning layers, it discovers potential spatiotemporal dependencies between nodes.

STGNN [29]: This model achieves the extraction of spatiotemporal information through a combination of GCN and GRU. Besides, it effectively aggregates information from surrounding roads using a learnable positional attention mechanism, thereby capturing both local and global spatiotemporal information present in the data.

Traditional optimization-based AGSTA: The principle is consistent with AGSTA, but the MAE loss function is adopted as the optimization approach.

Taking the 3-hour prediction as an example, the important parameter selections for deep learning models are shown in Table 4.

C. ERROR COMPARISON OF WIND FARM CLUSTER

All algorithms involved in this paper were tested in PyTorch 1.8.1 under the Linux system. The specific configuration of the server is as follows: the CPU is an Intel E5-2690V4 8-core processor, operating at a frequency of 2.6GHz and the GPU is Tes-V100, which is equipped with 32GB of video memory. The predictions were implemented using the sliding window prediction model with window sizes of 1h, 2h, 3h, 4h, 5h, and 6h, covering all the periods of the short-term prediction.

The performance of each model is shown in Table 5, where the data represents the prediction errors of the wind farm cluster's power.

1. In terms of single models: From the data, it is evident that SVR's predictive performance falls significantly behind deep learning methods within the 6-hour window. Especially as the time window expands, various indicators deteriorate sharply, further highlighting the model's poor predictive capabilities. This is mainly because superficial learning methods such as SVR have great limitations in dealing with non-linear relationships between high-dimensional time series and cannot learn the spatiotemporal characteristics of the data.

Compared to SVR, Deep Forecast demonstrates a noticeable improvement in predictive performance. This indicates that the stacking of LSTM layers can effectively extract temporal characteristics from the data and utilize these characteristics to predict data changes. However, in comparison to algorithms associated with GNNs, despite outperforming Graph WN in terms of WAPE for the 3-hour to 5-hour range, Deep Forecast still exhibits some limitations in overall predictive capability. This is because as the time window gradually increases, the temporal correlation of the data will gradually weaken. It is difficult to achieve accurate predictions relying solely on temporal relationships. At this time, the intervention of spatial relationships is needed to strengthen predictions.

2. In terms of hybrid models. From the data in the Table, it is evident that the predictive performance of IPDL is significantly better than SVR. This is because IPDL can capture unsupervised temporal features in the data by learning its probability distribution. Additionally, RAE's performance slightly surpasses IPDL, indicating that the seq2seq architecture, compared to simpler stacking, is better at learning temporal features from historical data. However, the performance of RAE in the 2-hour to 6-hour range falls behind that of Deep Forecast, implying that the architecture of this model lags in capturing temporal characteristics compared to LSTM.

TABLE 5. Performance comparison of AGSTA and other baseline models.

Time windows	Metrics	SVR	IPDL	RAE	Deep Forecast	DTDL	STGNN	Graph WN	MTGNN	T-AGSTA	AGSTA
1h	MAE	3.49	1.76	1.73	1.84	1.39	1.17	1.25	1.14	1.12	**1.09**
	RMSE	4.08	2.82	2.82	2.94	2.32	2.09	2.02	2.09	**1.95**	**1.95**
	WAPE (%)	64.58	31.54	31.36	33.07	24.88	21.06	22.53	**20.55**	22.13	22.07
2h	MAE	4.11	2.73	2.68	1.94	1.84	1.74	1.75	1.70	1.68	**1.64**
	RMSE	4.63	4.03	4.00	3.02	2.96	2.97	**2.71**	2.90	2.74	2.78
	WAPE (%)	69.68	48.86	47.83	34.67	34.42	31.05	31.21	**30.30**	32.78	32.69
3h	MAE	4.74	3.40	3.35	2.28	2.11	2.16	2.27	2.13	2.06	**2.01**
	RMSE	5.21	4.81	4.75	3.51	3.38	3.51	3.27	3.46	**3.22**	3.29
	WAPE (%)	76.74	60.51	59.55	40.18	37.53	38.38	40.40	**37.90**	39.84	38.15
4h	MAE	5.19	3.94	3.88	2.54	2.57	2.50	2.65	2.49	2.37	**2.31**
	RMSE	5.88	5.41	5.33	3.86	3.88	3.95	3.78	3.82	**3.63**	3.64
	WAPE (%)	82.37	69.81	68.81	45.02	48.04	**44.26**	46.94	48.12	46.69	44.58
5h	MAE	6.57	4.32	4.31	2.85	2.93	2.85	2.87	2.86	2.67	**2.63**
	RMSE	6.70	5.73	5.74	4.27	4.42	4.30	4.03	4.23	**4.03**	4.06
	WAPE (%)	90.48	76.23	76.10	50.35	51.65	50.24	50.62	54.35	50.48	**49.19**
6h	MAE	7.20	4.60	4.65	3.49	3.43	3.11	3.30	3.16	2.91	**2.80**
	RMSE	7.41	6.00	6.05	4.85	5.00	4.56	4.47	4.54	**4.30**	4.38
	WAPE (%)	97.10	80.96	81.84	61.45	60.43	54.81	58.14	59.73	54.83	**52.36**

In the 6-hour prediction range, the MAE and RMSE metrics of DTDL outperform the four methods mentioned earlier comprehensively, and even in the 2-hour to 4-hour prediction range, its performance is on par with models related to GNNs. The reasons for this advantage can be summarized in two aspects: firstly, due to the combination of LSTM in a seq2seq architecture, the model captures deeper levels of temporal features compared to the simpler LSTM stacking in Deep Forecast; secondly, the sparse coefficient vector can extract partial spatial information through the linear combination process with the original dictionary.

For the five GNN-related methods, due to their ability to deeply explore the spatiotemporal characteristics of the data, their overall performance surpasses the aforementioned models. Specifically, STGNN captures local spatiotemporal features in the data through the combination of GCN and GRU, and it captures global temporal features using attention mechanisms. Through this approach, prominent predictive effects can be achieved, particularly when the prediction window is 4 hours, where its WAPE metric stands out as the best among all models. On the other hand, Graph WN and MTGNN share structural similarities, with the difference that the former integrates DC layers into the stacked CNN structure, while the latter improves the combination of certain CNN components, forming GC and TC layers. This leads to their respective advantages in predictive performance, Graph WN outperforms MTGNN in terms of RMSE, while MTGNN surpasses Graph WN in MAE. Moreover, MTGNN achieves the best WAPE metric within the 3-hour window among all algorithms.

However, these three GNN-related models mentioned above, while all focused on extracting spatiotemporal characteristics, overlooked the spatiotemporal coupling relationships within wind power data. As a result, their overall performance falls behind AGSTA, a model equipped with a spatiotemporal attention computation module. This module effectively calculates deeper-level information within the spatiotemporal characteristics of the data, enabling more accurate predictions of multi-site wind power. More specifically, STGNN performs attention computation after extracting local spatiotemporal information. The whole process emphasizes the calculation of temporal relationships rather than the spatial dependencies among different sites. The algorithmic structures of Graph WN and MTGNN are based on combinations of CNN, which are limited in describing spatiotemporal dependencies across multiple wind farms in a non-Euclidean space.

Comparing AGSTA and T-AGSTA, the advantage of T-AGSTA with MAE as the loss function is evident in the RMSE metric after 2 hours. However, in the 6-hour prediction range, AGSTA outperforms T-AGSTA in terms of both MAE and WAPE. This directly indicates that the hybrid loss function is more suitable for enhancing the predictive accuracy of multiple sites over multiple time steps, as opposed to the traditional MAE loss function.

Figure 9 presents a box plot of the errors between predicted values and actual values for different models under a 3-hour prediction horizon. To provide a clearer representation, the plot excludes outliers.

Regarding the interquartile range (IQR), from the box plot, it can be observed that Deep Forecast, RAE, IPDL, and

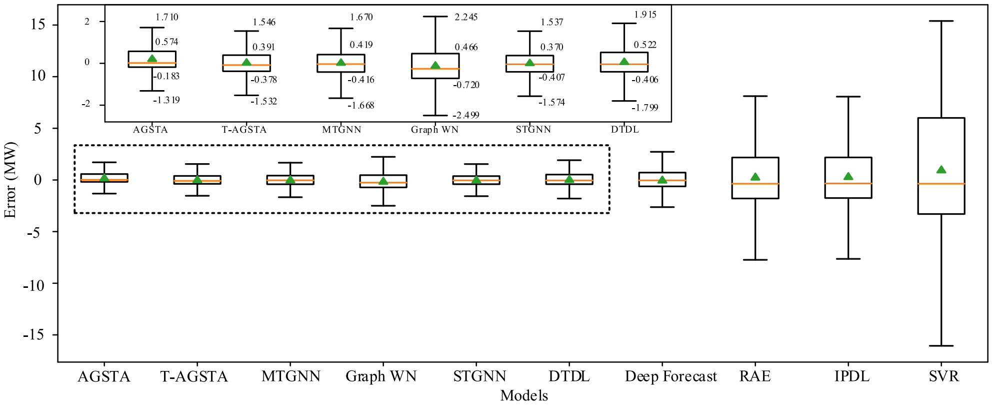
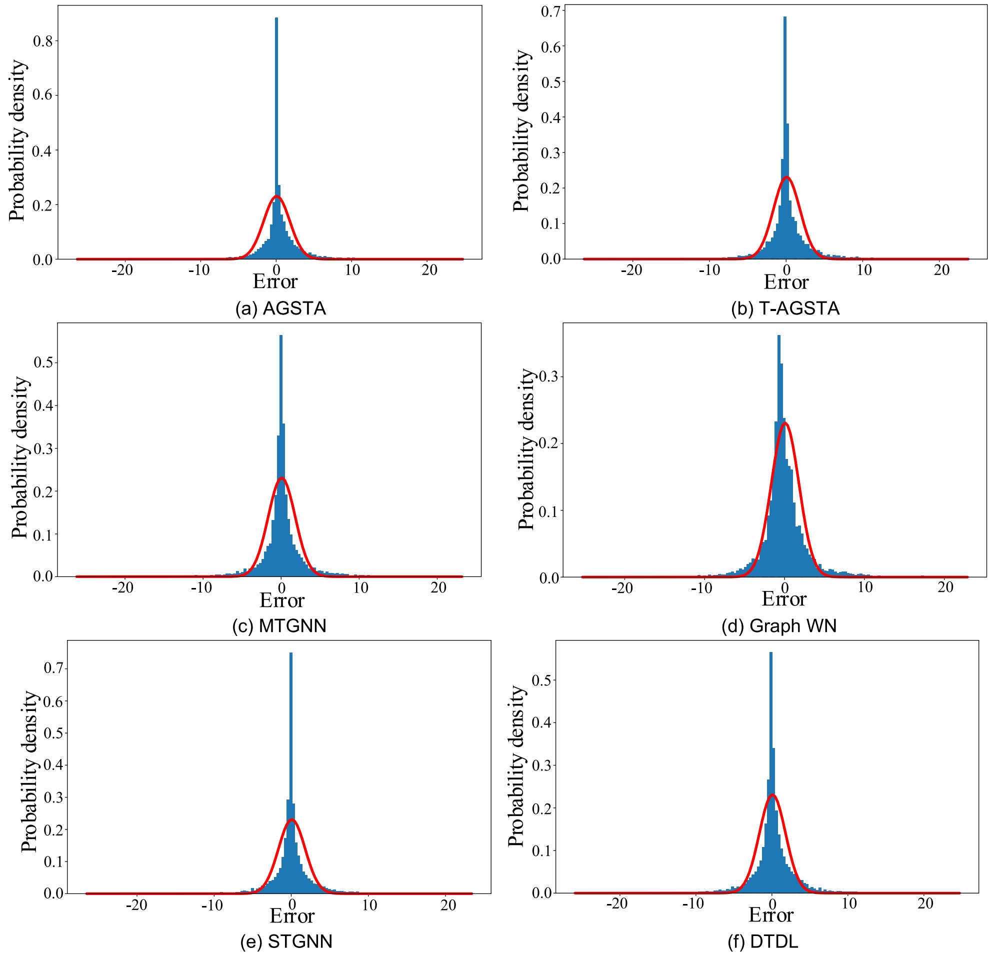


FIGURE 9. Box plot comparison of prediction errors among different models.



(a) AGSTA
(b) T-AGSTA
(c) MTGNN
(d) Graph WN
(e) STGNN
(f) DTDL

FIGURE 10. Error probability density distribution.

SVR exhibit significantly larger IQRs, indicating broader and more volatile error distributions for these four algorithms. Meanwhile, DTDL, Graph WN, MTGNN, STGNN, T- AGSTA, and AGSTA have IQRs of 0.928, 1.186, 0.835, 0.777, 0.769, and 0.757 respectively, highlighting AGSTA's more concentrated error distribution. For the median line, corresponding to the order of the scaled graph, the median values for various models are 0.017, −0.081, −0.036, −0.258, −0.043, and −0.043. AGSTA's median is closest to 0, indicating its error distribution is closer to a uniform distribution.

TABLE 6. MAE comparison of different models for each wind farm.

Wind Farm	SVR	IPDL	RAE	Deep Forecast	DTDL	STGNN	Graph WN	MTGNN	T-AGSTA	AGSTA
WF1	3.68	3.49	3.44	2.59	2.49	2.22	2.33	2.23	**2.18**	2.20
WF2	4.10	3.87	3.86	2.27	2.34	2.18	2.28	2.28	2.08	**2.01**
WF3	4.28	3.86	3.85	2.27	2.39	2.22	2.27	2.27	2.03	**2.02**
WF4	4.76	3.23	3.16	2.69	2.48	2.41	2.39	2.39	2.24	**2.18**
WF5	5.01	3.15	3.14	2.31	2.10	2.12	2.23	2.23	2.03	**1.98**
WF6	4.96	3.08	3.11	2.89	2.46	2.40	2.64	2.64	2.17	**2.06**

TABLE 7. RMSE comparison of different models for each wind farm.

Wind Farm	SVR	IPDL	RAE	Deep Forecast	DTDL	STGNN	Graph WN	MTGNN	T-AGSTA	AGSTA
WF1	4.94	4.65	3.65	3.69	3.71	3.47	**3.31**	3.45	3.36	3.40
WF2	5.01	5.11	5.10	3.37	3.54	3.38	3.17	3.42	**3.10**	3.16
WF3	4.98	5.07	4.57	3.38	3.52	3.50	3.22	3.44	**3.11**	3.16
WF4	5.25	4.49	4.40	3.91	3.72	3.68	3.40	3.55	**3.33**	3.44
WF5	5.33	4.56	4.56	3.50	3.36	3.37	3.26	3.29	**3.08**	3.09
WF6	5.56	4.82	4.86	4.49	4.07	3.90	3.88	3.73	**3.37**	3.42

TABLE 8. WAPE (%) comparison of different models for each wind farm.

Wind Farm	SVR	IPDL	RAE	Deep Forecast	DTDL	STGNN	Graph WN	MTGNN	T-AGSTA	AGSTA
WF1	67.69	59.18	58.28	43.97	41.95	**37.59**	39.54	40.38	41.08	40.75
WF2	55.21	52.05	51.98	30.52	31.57	**29.30**	32.28	31.17	30.61	30.46
WF3	57.68	53.05	53.02	31.28	32.39	**30.62**	31.51	32.20	31.15	31.30
WF4	81.24	76.65	79.84	67.96	61.50	56.64	60.48	61.06	56.47	**56.38**
WF5	60.02	54.90	36.40	40.30	36.40	**36.98**	39.18	39.89	41.43	40.67
WF6	86.71	86.60	87.43	81.02	68.70	67.50	74.40	67.22	45.87	**44.62**

Combining the above analysis, in the context of a 3-hour prediction, AGSTA exhibits the best overall predictive performance, aligning with the conclusions presented in Table 4.

In order to provide a more comprehensive display of the distribution of errors, Figure 10 shows the probability density distribution of prediction errors for six models in the scaled portion of Figure 9.

D. ERROR COMPARISON OF EACH WIND FARM

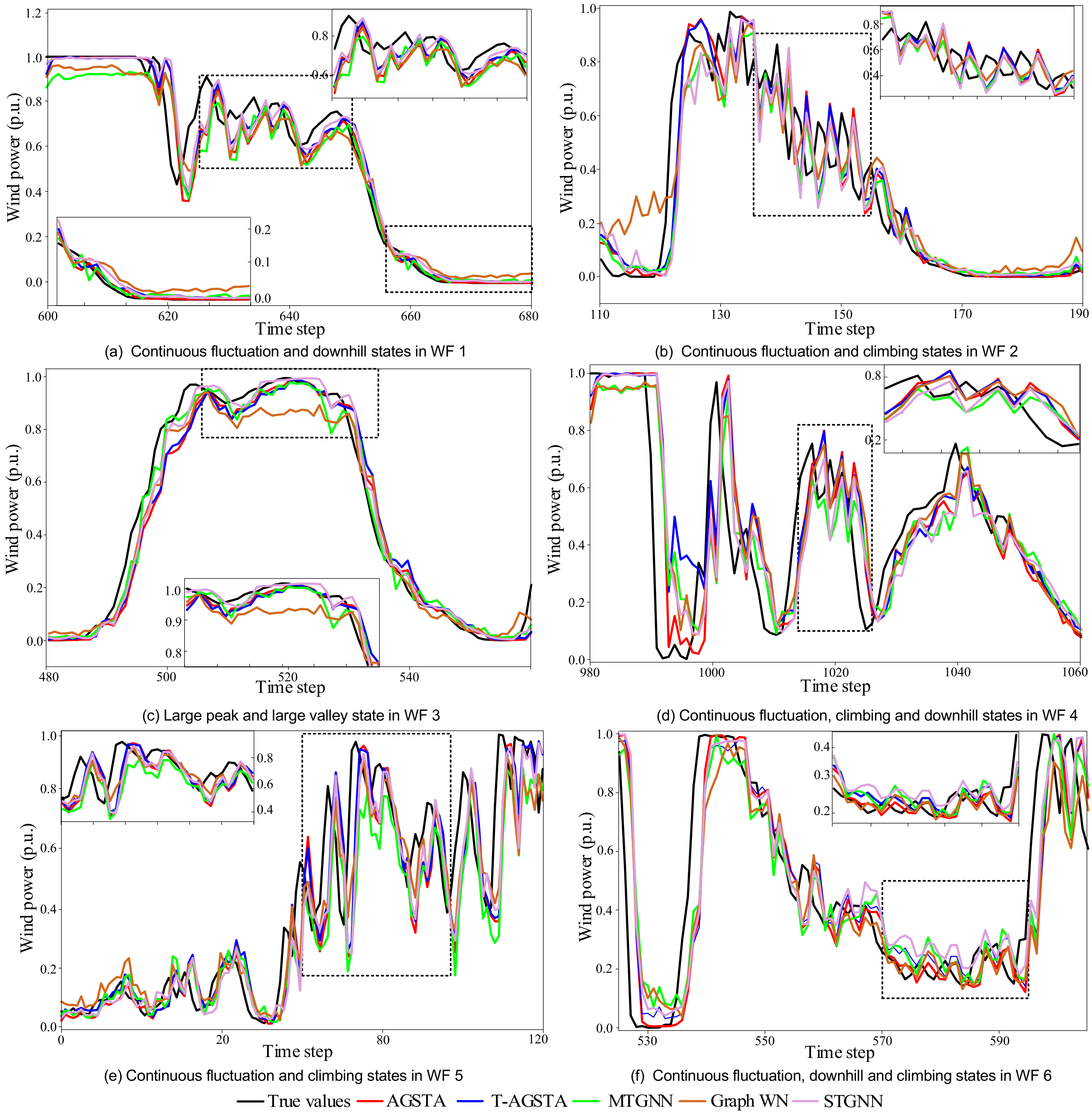
To provide a clearer reflection of the power prediction for each wind farm, Tables 6 to 8 present the average errors for each wind farm across the six-time windows.

From the error data presented in the following three tables, it can be observed that, except for AGSTA and T-AGSTA, the remaining models exhibit notably higher errors when predicting the power of WF4 and WF6 compared to the other sites. This is because the power output of these two wind farms is significantly lower than the other four sites, making it difficult for the model to adequately consider these two sites when extracting the spatial characteristics between adjacent sites. However, AGSTA utilizes spatiotemporal attention calculation to comprehensively assess the coupled relationships existing in both time and space within the data. This allows the model to simultaneously consider the predictions for each individual wind farm while performing the wind farm cluster power prediction. From the data in the above three Tables, it is evident that the model based on spatiotemporal coupling calculation yields the best performance in predicting WF4 and WF6.

More specifically, T-AGSTA performs the best in terms of the RMSE metric, indicating that the predictions obtained using traditional loss functions have better stability. STGNN holds an overall advantage in the WAPE metric but exhibits poorer predictions for WF4 and WF6. AGSTA outperforms other models in terms of overall performance based on the MAE metric, as well as on the WAPE metric for predicting WF4 and WF5.

E. COMPARISON OF LOCAL PREDICTION CURVES

Figure 10 shows a comparison of the local power prediction results of five GNN-related algorithms under a 3h prediction window. These power curves include four common states in wind power curves, which are continuous fluctuation state, climbing state, downhill state, and large peak and large valley state.



(a) Continuous fluctuation and downhill states in WF 1

(b) Continuous fluctuation and climbing states in WF 2

(c) Large peak and large valley state in WF 3

(d) Continuous fluctuation, climbing and downhill states in WF 4

(e) Continuous fluctuation and climbing states in WF 5

(f) Continuous fluctuation, downhill and climbing states in WF 6

FIGURE 11. Prediction power curves of 6 wind farms.

It can be observed that except for the large peak and large valley in (c) where MTGNN holds an advantage, in all other scenarios, whether during the climbing stage, downhill state, or continuous fluctuation state, the comprehensive performance of power curve prediction by AGSTA and T-AGSTA is notably superior to the rest of the algorithms. Scaling portions in (a), (b), and (e) reveal that during continuous rapid fluctuation states, the performance of T-AGSTA can rival that of AGSTA. However, in states where the power curve is relatively gentle, as depicted in the scaled portions of (a), (c), and (f), AGSTA exhibits a significant advantage.

F. TIME COMPLEXITY

The structures of AGSTA and T-AGSTA are identical, with their core performing parallel spatiotemporal attention calculation. Their time complexity can be roughly represented as $\mathcal{O}((E_{\max} + V_{\max})T^2)$. Here, T represents the temporal length of the data, and $E_{\max}$ and $V_{\max}$ are the quantities of edges and nodes, respectively, in the graph constructed based on the temporal dimension of the data for executing spatiotemporal attention calculation.

MTGNN primarily consists of a combination of CNN with multiple different kernel sizes, making it challenging to provide a comprehensive estimation of its time complexity. Thus, only the time complexities for TC and GC are presented here. The time complexity for TC is $\mathcal{O}(k_1 TV^2 + k_2 TV^2 + k_3 TV^3 + k_4 TV^2)$, while for GC, it is $\mathcal{O}(k_5 TV^2)$, the sum of these two can be roughly considered as the time complexity of MTGNN. Here, k_1, k_2, k_3, k_4, and k_5 represent the sizes of convolutional kernels in different convolutions, and V signifies the number of nodes in the adaptive graph. Similarly, for Graph WN, only the time complexity of DC is given as $\mathcal{O}(KE)$, and its time complexity can be viewed as the sum of multiple convolutions and DC. Here, K denotes the number of diffusion steps, and E represents the quantity of edges in the adaptive graph.

The time complexity of STGNN is composed of three parts: the time complexity of GCN is $\mathcal{O}(E)$, the time complexity of the combination of GRU and GCN is $\mathcal{O}\left(TV^2 + E\right)$, and the time complexity of the self-attention mechanism is $\mathcal{O}(T^2V)$. Therefore, the final time complexity is $\mathcal{O}(E{+}(V + T)TV)$.

VII. CONCLUSION

This paper presents a novel short-term power prediction method. Firstly, the deficiencies of traditional static graphs in reflecting the spatiotemporal relationships between nodes are demonstrated, and an adaptive undirected graph is constructed to replace a static graph. Next, by utilizing cross-attention and self-attention within the spatiotemporal attention layer, the coupling relationships between data and temporal dependencies are computed, respectively. Subsequently, a hybrid loss function is proposed for accurate optimization. Finally, through validation, it is demonstrated that, among the selected comparative algorithms, the proposed method exhibits better performance in wind power prediction accuracy. The detailed conclusions of this paper are as follows:

(1) Compared to static graphs, the adaptive undirected graph constructed in this paper is more suitable for reflecting the time-varying relationships among nodes arising from dynamic spatiotemporal correlations.

(2) Through iterative operations of the ST layer, the coupling relationships between data and temporal dependencies can be effectively computed. Besides, the repetitive execution of message passing ensures that information aggregation and propagation are not limited to neighboring nodes. This design significantly outperforms other models selected in this paper in terms of predictive performance.

(3) Compared to the traditional MAE loss function, the hybrid loss function proposed in this paper shows better comprehensive performance in predictive accuracy.

REFERENCES

[1] C. Croonenbroeck and G. Stadtmann, "Renewable generation forecast studies—Review and good practice guidance," *Renew. Sustain. Energy Rev.*, vol. 108, pp. 312–322, Jul. 2019.

[2] Global Wind Energy Council, São Paulo, Brazil. *Global Wind Report 2023*. Accessed: Sep. 26, 2023. [Online]. Available: https://gwec.net/globalwindreport2023

[3] A. Meng, J. Ge, H. Yin, and S. Chen, "Wind speed forecasting based on wavelet packet decomposition and artificial neural networks trained by crisscross optimization algorithm," *Energy Convers. Manage.*, vol. 114, pp. 75–88, Apr. 2016.

[4] Y. Song, D. Tang, J. Yu, Z. Yu, and X. Li, "Short-term forecasting based on graph convolution networks and multiresolution convolution neural networks for wind power," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1691–1702, Feb. 2023.

[5] V. Prema, M. S. Bhaskar, D. Almakhles, N. Gowtham, and K. U. Rao, "Critical review of data, models and performance metrics for wind and solar power forecast," *IEEE Access*, vol. 10, pp. 667–688, 2022.

[6] I. Marisca, A. Cini, and C. Alippi, "Learning to reconstruct missing data from spatiotemporal graphs with sparse observations," 2022, *arXiv:2205.13479*.

[7] S. Lu, "Multi-step ahead ultra-short-term wind power forecasting based on time series analysis," in *Proc. Int. Conf. Comput. Inf. Big Data Appl. (CIBDA)*, Guiyang, China, Apr. 2020, pp. 430–434.

[8] E. Erdem and J. Shi, "ARMA based approaches for forecasting the tuple of wind speed and direction," *Appl. Energy*, vol. 88, no. 4, pp. 1405–1414, Apr. 2011.

[9] Aasim, S. N. Singh, and A. Mohapatra, "Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting," *Renew. Energy*, vol. 136, pp. 758–768, Jun. 2019.

[10] P. Jiang, Z. Liu, X. Niu, and L. Zhang, "A combined forecasting system based on statistical method, artificial neural networks, and deep learning methods for short-term wind speed forecasting," *Energy*, vol. 217, Feb. 2021, Art. no. 119361.

[11] P. Jiang, Y. Wang, and J. Wang, "Short-term wind speed forecasting using a hybrid model," *Energy*, vol. 119, pp. 561–577, Jan. 2017.

[12] V. Cherkassky and Y. Ma, "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Netw.*, vol. 17, no. 1, pp. 113–126, Jan. 2004.

[13] Y. Zhang, P. Wang, C. Zhang, and S. Lei, "Wind energy prediction with LS-SVM based on Lorenz perturbation," *J. Eng.*, vol. 2017, no. 13, pp. 1724–1727, Jan. 2017.

[14] M. Khodayar, O. Kaynak, and M. E. Khodayar, "Rough deep neural architecture for short-term wind speed forecasting," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 2770–2779, Dec. 2017.

[15] M. Khodayar, J. Wang, and M. Manthouri, "Interval deep generative neural network for wind speed forecasting," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3974–3989, Jul. 2019.

[16] A. Ghaderi, B. M. Sanandaji, and F. Ghaderi, "Deep forecast: Deep learning-based spatio-temporal forecasting," 2017, *arXiv:1707.08110*.

[17] C. Li, G. Tang, X. Xue, A. Saeed, and X. Hu, "Short-term wind speed interval prediction based on ensemble GRU model," *IEEE Trans. Sustain. Energy*, vol. 11, no. 3, pp. 1370–1380, Jul. 2020.

[18] M. Khodayar, J. Wang, and Z. Wang, "Energy disaggregation via deep temporal dictionary learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1696–1709, May 2020.

[19] H. Acikgoz, "A novel approach based on integration of convolutional neural networks and deep feature selection for short-term solar radiation forecasting," *Appl. Energy*, vol. 305, Jan. 2022, Art. no. 117912.

[20] G. Marco, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 2, 2005, pp. 729–734.

[21] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[22] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. Int. Conf. Learn. Represent.*, 2013, pp. 1–14.

[23] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–16.

[24] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1907–1913.

[25] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 753–763.

[26] Z. Li, L. Ye, Y. Zhao, M. Pei, P. Lu, Y. Li, and B. Dai, "A spatiotemporal directed graph convolution network for ultra-short-term wind power prediction," *IEEE Trans. Sustain. Energy*, vol. 14, no. 1, pp. 39–54, Jan. 2023.

[27] F. Wang, P. Chen, Z. Zhen, R. Yin, C. Cao, Y. Zhang, and N. Duić, "Dynamic spatio-temporal correlation and hierarchical directed graph structure based ultra-short-term wind farm cluster power forecasting method," *Appl. Energy*, vol. 323, Oct. 2022, Art. no. 119579.

[28] Y. Liu, Q. Liu, J.-W. Zhang, H. Feng, Z. Wang, Z. Zhou, and W. Chen, "Multivariate time-series forecasting with temporal polynomial graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 19414–19426.

[29] X. Geng, L. Xu, X. He, and J. Yu, "Graph optimization neural network with spatio-temporal correlation learning for multi-node offshore wind speed forecasting," *Renew. Energy*, vol. 180, pp. 1014–1025, Dec. 2021.

[30] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, "Traffic flow prediction via spatial temporal graph neural network," in *Proc. Web Conf.*, New York, NY, USA, Apr. 2020, pp. 1082–1092.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.

[32] X. Kong, W. Xing, X. Wei, P. Bao, J. Zhang, and W. Lu, "STGAT: Spatial-temporal graph attention networks for traffic flow forecasting," *IEEE Access*, vol. 8, pp. 134363–134372, 2020.

[33] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[34] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–16.

[35] C. Draxl, A. Clifton, B.-M. Hodge, and J. McCaa, "The wind integration national dataset (WIND) toolkit," *Appl. Energy*, vol. 151, pp. 355–366, Aug. 2015.

HAOCHEN LI received the B.E. and M.E. degrees from the Taiyuan University of Science and Technology, Taiyuan, China, in 2011 and 2016, respectively, where he is currently pursuing the Ph.D. degree. He is a Teacher with the Shanxi Institute of Technology, Yangquan, China. His current research interests include the data mining of power systems and the control of microgrid systems.

LIQUN LIU received the Ph.D. degree from the Department of Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2011. He is currently a Professor with the Department of Electronic and Information Engineering, Taiyuan University of Science and Technology, Taiyuan, China. His current research interests include the control of renewable energy systems, optimization configuration of power supply systems, electrical supplies, and high-efficiency power electronic converters.

QIUSHENG HE received the Ph.D. degree from the School of Mechanical and Electrical Engineering, China University of Mining and Technology, Beijing, China, in 2007. He is currently a Professor with the Department of Electronic and Information Engineering, Taiyuan University of Science and Technology, Taiyuan, China. His current research interests include robot control and machine vision.