Short-term Wind Power Cluster Prediction Method Based on Multi-Spatial-Scale Features

# Short-term Wind Power Cluster Prediction Method Based on Multi-Spatial-Scale Features


1st Mao Yang
*Key Laboratory of Modern Power System Simulation and Control & Renewable Energy Technology，Ministry of Education (Northeast Electric Power University)*
Jilin 132012, China
E-mail:yangmao820@163.com

2nd Bozhi Dai
*Key Laboratory of Modern Power System Simulation and Control & Renewable Energy Technology，Ministry of Education (Northeast Electric Power University)*
*Zhongshan Power Supply Bureau of Guangdong Power Grid Co., Ltd.*
Jilin 132012, China
E-mail:daibozhi123@163.com

3rd Zhiyuan Ma
*Inner Mongolia East Electric Power Co., Ltd. Electric Power Science Research Institute*
Jilin 132012, China
E-mail:450102025@qq.com

4th Jingsi Chen
*Department of Energy, Politecnico di Torino, Torino 10129, Italy.*
E-mail: jingsi.chen@polito.it

5th Jun Yin
*Department of Electronics and Telecommunications, Politecnico di Torino, Torino 10129, Italy.*
E-mail: jun.yin@polito.it



***Abstract*—To address the high modeling complexity in wind power cluster prediction caused by wide-area high-dimensional meteorological features, this study builds feature extraction modules at global and wind farm scales. Specifically, it uses Global Block-wise Attention Transformer for global feature processing and Dynamic Hybrid Sparse Graph Attention for wind farm-level feature extraction to reduce model complexity and improve computational efficiency. Applied to a Jilin wind power cluster, results show the method effectively enhances model computational efficiency and wind power output prediction accuracy.**



***Keywords—component, wind power cluster, power prediction, feature extraction, transformer, graph attention***


## I. INTRODUCTION

With the annual increase in the proportion of wind power in the power grid, high-precision wind power prediction has become an essential means to ensure the safe and stable operation of power systems [1]. Among various types of wind power prediction, short-term wind power prediction is applied to optimize daily power generation plans and adjust maintenance plans, which holds significant importance for power systems [2].In recent years, the rapid growth in the number of wind farms and the continuous expansion of wind power cluster scales have led to a significant increase in the dimensionality of Numerical Weather Prediction (NWP) data, which serves as the core input for prediction models. This not only raises the training costs of prediction models but also increases their training complexity. However, directly feeding wide-area high-dimensional meteorological features into prediction models will significantly escalate the models' computational and storage demands. This issue is particularly prominent for complex architectures represented by Transformers and graph neural networks.

In addition to the challenge of computational efficiency, how to efficiently mine and utilize meteorological features on a large spatial scale has also become a crucial issue in wind power prediction. Reference [3] constructs a graph convolutional neural network model to extract the associated features of meteorological map nodes from non-Euclidean space, and feeds them into a gated recurrent unit network integrated with an attention mechanism to enhance the ability of temporal feature extraction. Reference [4] builds a wind power cluster prediction model using a residual network based on grid-based NWP, aiming to mine the correlation between meteorological distribution and power output. Reference [5] develops a multi-output prediction model for regions using a residual network, which fully explores the associations between the distribution of regional meteorological resources and the power generation of individual wind farms. However, although the aforementioned methods use grid-based NWP and take overall meteorological features as input, they lack personalized modeling for the regional and local meteorological conditions of wind farms, which limits the prediction performance of the models.

Therefore, this paper divides NWP data into two distinct spatial scales: the global scale and the node scale. Specifically, the global scale corresponds to the complete NWP dataset, and a global feature extraction module is constructed through data decomposition, recombination, and the introduction of global variables. The node scale corresponds to NWP data at the locations of individual wind farms, and a node feature extraction module is built based on Dynamic Hybrid Sparse Graph Attention (DHS-GAT). Finally, short-term wind power cluster prediction is performed under the features of different spatial scales, followed by a comparative analysis.

## II. METHODOLOGY

### A. Node Feature Importance Matrix

The input data is $\boldsymbol{X} \in \boldsymbol{R}^{T\ F\ H\ W}$ , where $\boldsymbol{R}$ represents a four-dimensional grid-type NWP dataset. In this notation, $T$ denotes the time length of the sample; $F$ denotes the number of feature channels; $H$ and $W$ correspond to latitude and longitude, respectively; and the spatial resolution of NWP is $0.1 \quad 0.1$ .

From a smaller spatial scale perspective, influenced by topographic and meteorological factors, nodes with relatively close spatial distances exhibit highly similar meteorological features, as illustrated in Fig. 1.

At a larger spatial scale, the Pearson correlation coefficient matrix $\boldsymbol{Z} \in \boldsymbol{R}^{H \times W}$ between the total power of a wind farm cluster in Jilin Province and the wind speed of grid-type NWP is calculated, and the formula is as follows:


Smart Grid-National Science and Technology Major Project (2025ZD0805500).

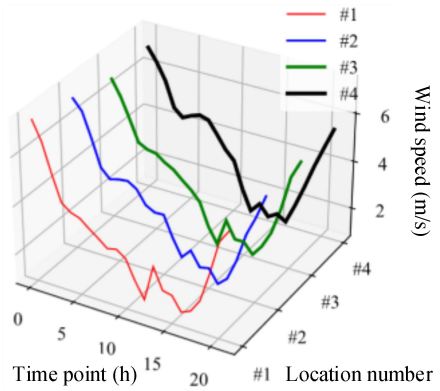


Fig. 1. Wind speeds at the near neighbor nodes

$$\boldsymbol{Z}_{h,w}=\frac{\sum_{t=1}^{T}\left(S_{h,w}(t)-\overline{S}_{h,w}\right)\left(P_{\text{sum}}(t)-\overline{P}_{\text{sum}}\right)}{\sqrt{\sum_{t=1}^{T}\left(S_{h,w}(t)-\overline{S}_{h,w}\right)^{2}}\sqrt{\sum_{t=1}^{T}\left(P_{\text{sum}}(t)-\overline{P}_{\text{sum}}\right)^{2}}} \tag{1}$$

In the formula, $\boldsymbol{Z}_{h,w}$ is the element of $\boldsymbol{Z}$ at position $(h,w)$; $S_{h,w}(t)$ denotes the wind speed at location $(h,w)$ at time $t$; $\overline{S}_{h,w}$ is the mean wind speed at location $(h,w)$; $P_{\text{sum}}(t)$ represents the total cluster power at time $t$; and $\overline{P}_{\text{sum}}$ is the mean total cluster power.

Matrix $\boldsymbol{Z}$ is visualized as a correlation heatmap, with the locations of wind farms annotated, as depicted in Fig. 2. It can be clearly observed from the figure that the correlation is significantly influenced by the distribution of wind farms; generally, the denser the wind farms are distributed, the stronger the correlation of meteorological features tends to be.

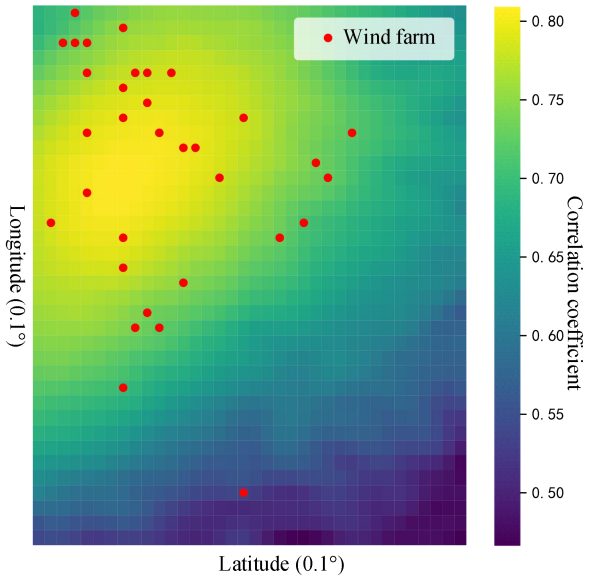


Fig. 2. Correlation of total cluster power with NWP wind speeds

High correlation usually indicates that the meteorological feature importance of the node is higher. Taking such feature importance as prior information and inputting it into the prediction model can reduce the interference of irrelevant features on model training, guide the model to focus on key features, and accelerate model convergence [6]. Thus, this paper refers to the Softmax function to calculate the node feature importance matrix $\boldsymbol{Z}_{\text{g}}$, which is used to weight NWP. The calculation formula is as follows:

$$\boldsymbol{Z}_{\text{g}}=\frac{\exp\left(\boldsymbol{Z}_{h,w}\right)}{\sum_{h=1}^{H}\sum_{w=1}^{W}\exp\left(\boldsymbol{Z}_{h,w}\right)} \tag{2}$$

In the formula, exp (·) denotes the natural exponential function.

### *B. Feature Extraction at Different Spatial Scales*

To capture information of data at different levels and more accurately describe the relationship between complex meteorological features and wind power output, this study designs feature extraction modules at two scales: the global scale and the node scale. The self-attention mechanism is a widely used deep learning modeling method. It enables the model to dynamically focus on other elements in the same input sequence when processing a specific input element, thereby capturing the relationships between different parts [7].

For the NWP data $\boldsymbol{X}$ input to the model, the attention scores are calculated via linear data transformation using the following formula:

$$\boldsymbol{\Theta}=\partial\left(\frac{\left(\boldsymbol{X}\bullet\boldsymbol{U}_{Q}\right)\left(\boldsymbol{X}\bullet\boldsymbol{U}_{K}\right)^{\text{T}}}{\sqrt{d_k}}\right) \tag{3}$$

In the formula, $\boldsymbol{U}_{Q}$ and $\boldsymbol{U}_{K}$ denote linear transformation matrices; $d_k$ represents the scaling factor for the Softmax activation function, which prevents potential gradient vanishing caused by excessively large values.

Since the NWP data is $\boldsymbol{X}\in\boldsymbol{R}^{N\ F\ H\ W}$, its self-attention complexity is $O\left(H^{2}\ \ W^{2}\ \ d_k\right)$ [8]. As the resolution and spatial scale increase, the complexity grows quadratically, significantly increasing the computational load and memory requirement during feature extraction.

### *C. Global Variables*

To reduce the training search space and improve training efficiency and convergence speed, this paper multiplies the NWP $\boldsymbol{X}$ element-wise with the node feature importance matrix $\boldsymbol{Z}_{\text{g}}$ to obtain the weighted NWP $\boldsymbol{X}_{\text{g}}$. Then, $\boldsymbol{X}_{\text{g}}$ is decomposed into $n$ non-overlapping data blocks $\mathbf{x}^{(n)}$ along the spatial dimension.

However, decomposing NWP into small data blocks to calculate attention scores independently effectively reduces the computational complexity, but each data block can only perceive the information within the block and cannot capture the global correlations between blocks. Therefore, a global variable $\boldsymbol{\Omega}\in\boldsymbol{R}^{\lambda\times F}$ is introduced, where λ is a user-defined hyperparameter. $\boldsymbol{\Omega}$ enables each data block, when performing the self-attention mechanism, to not only focus on elements

within the same data block but also on global vectors. Within each data block, the attention mechanism is used in parallel. After the self-attention transformation, the $n$-th data block $\boldsymbol{x}_{\text{attn}}^{(n)}$ with the introduction of the global vector is given by:

$$\begin{aligned}\boldsymbol{x}_{\text{attn}}^{(n)} &= \partial\left(\frac{\left(\boldsymbol{x}_Q^{(n)}\boldsymbol{U}_Q^{\Psi}\right)\left(\boldsymbol{x}_K^{(n)}\boldsymbol{U}_K^{\Psi}\right)^{\mathrm{T}}}{\sqrt{d_k}}\right)\left(\boldsymbol{x}_V^{(n)}\boldsymbol{U}_P^{\Psi}\right)\\ \boldsymbol{x}_Q^{(n)} &= \boldsymbol{x}^{(n)}\\ \boldsymbol{x}_K^{(n)} &= \boldsymbol{x}_V^{(n)} = \oplus\left(\boldsymbol{x}^{(n)}, \boldsymbol{\Omega}\right)\end{aligned} \tag{4}$$

In the formula, $\boldsymbol{U}_Q^{\Psi}$, $\boldsymbol{U}_K^{\Psi}$, and $\boldsymbol{U}_P^{\Psi}$ are linear transformation matrices. These three parameters are shared across all data blocks. $\boldsymbol{x}_Q^{(n)}$, $\boldsymbol{x}_K^{(n)}$, and $\boldsymbol{x}_V^{(n)}$ are the input matrices after introducing the global variable, respectively. $\oplus(\bullet)$ denotes matrix concatenation.

Finally, the transformed data of each block is restored to its original position to obtain the final main channel output $\boldsymbol{X}_{\text{out}}^{\text{global}}$ of the module and the output $\boldsymbol{\Omega}_{\text{out}}$ of the global variable. This paper refers to this model as Global Block-wise Attention Transformer (GBA-Transformer), and its structure is shown in Fig. 3.

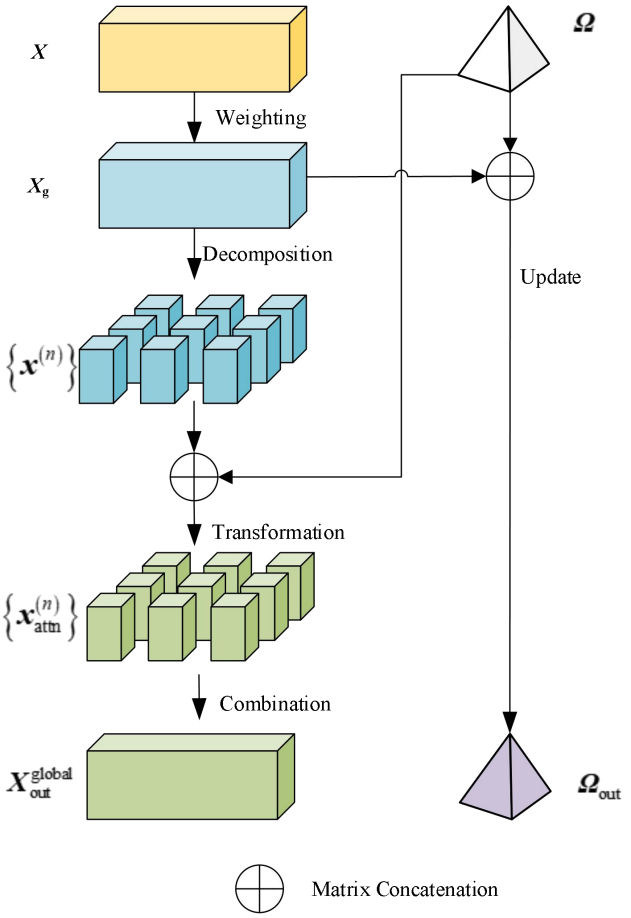


Fig. 3. Global Block-wise Attention Transformer

*D. Node Feature Extraction Module*

To capture meteorological features in key regions and provide accurate regional information support for power prediction, 40 wind farms within the region are divided into 3 clusters using the K-Means algorithm based on their spatial distances. The convex hull algorithm is applied to each cluster to map the shape of the wind farm clusters [9], as shown in Fig. 4.

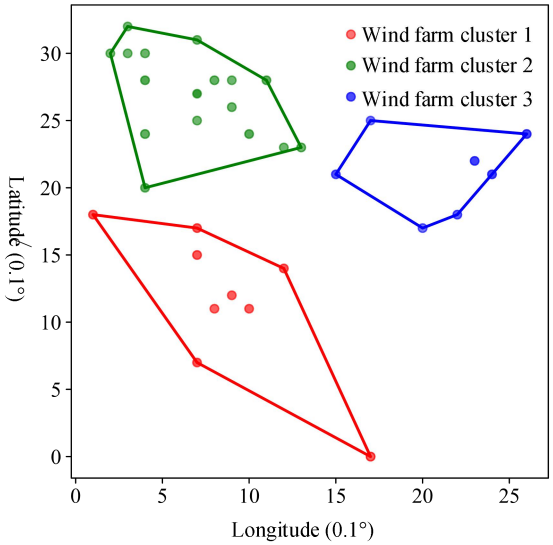


Fig. 4. Wind farm clusters and convex hulls

A regional mask $\boldsymbol{M}^i \in \boldsymbol{R}^{H\times W}$ is defined: for nodes inside and on the boundary of convex hull $i$, it is defined as 1; otherwise, it is defined as 0. Its calculation formula is as follows:

$$M_{h,w}^i = \begin{cases}1, & (h,w) \in D_i\\ 0, & (h,w) \notin D_i\end{cases} \tag{5}$$

In the formula, $D_i$ represents the $i$-th convex hull, where $i$=1,2,3.

GAT is a graph-structure-based deep learning method. It introduces an attention mechanism to dynamically assign weights to the relationships between different nodes in the graph [10]. Since the distribution of wind farms inherently exhibits graph properties, and the relationships between nodes can be represented by edges [11], this paper selects the GAT model for regional feature extraction. Each cluster is treated as one node in the graph structure, and the calculation formula for the feature vector $\boldsymbol{G}^i$ of cluster node $i$ is as follows:

$$\boldsymbol{G}^i = \left(\mathrm{PCA}\left(\oplus\left\{\boldsymbol{X}\odot\boldsymbol{Z}_{\mathrm{g}}^{*} \mid M_{h,w}^i = 1\right\}\right)\right) \tag{6}$$

In the formula, $\mathrm{PCA}(\bullet)$ represents Principal Component Analysis [12]; $\boldsymbol{X}$ still denotes NWP; $\boldsymbol{Z}_{\mathrm{g}}^{*}$ is the node feature importance matrix after dimension elevation.

The calculation formula of $\boldsymbol{Z}_{\mathrm{g}}^{*}$ is as follows:

$$\mathbf{Z}_g^* = \mathrm{Expand}\left(\mathbf{Z}_g, (1,1,H,W)\right) \tag{7}$$

In the formula, $\mathrm{Expand}(\bullet)$ denotes matrix dimension elevation.

The two-dimensional matrix is transformed into a four-dimensional matrix to make the dimension of the importance matrix match that of the input data.

The key to graph attention lies in calculating the attention weights between nodes. The formula for the attention weight $\alpha_{i,j}$ between cluster node *i* and its adjacent cluster node *j* is as follows:

$$\alpha_{i,j} = \frac{\exp\left(\varsigma\left(\boldsymbol{a}^{\mathrm{T}} \oplus \left(\boldsymbol{U}\boldsymbol{G}_i, \boldsymbol{U}\boldsymbol{G}_j\right)\right)\right)}{\sum_{k \in \Lambda(i)} \exp\left(\varsigma\left(\boldsymbol{a}^{\mathrm{T}} \oplus \left(\boldsymbol{U}\boldsymbol{G}_i, \boldsymbol{U}\boldsymbol{G}_k\right)\right)\right)} \tag{8}$$

In the formula, $\boldsymbol{a}$ is an attention parameter vector, used to calculate the attention score between nodes, $\boldsymbol{U}$ is a weight matrix; $\boldsymbol{G}_i$ and $\boldsymbol{G}_j$ are the feature vectors of cluster nodes *i* and *j*, respectively; $\Lambda(i)$ is the set of adjacent nodes of cluster node *i*, $\varsigma(\bullet)$ is the LeakyReLU activation function.

After graph attention weights the features of adjacent nodes, the final output $G_i^*$ of cluster node *i* is obtained by the following formula:

$$G_i^* = \sigma\left| \sum_{j \in \Lambda(i)} \alpha_{i,j} \boldsymbol{U}\boldsymbol{G}_j \right| \tag{9}$$

In the formula, $\sigma(\cdot)$ is a nonlinear activation function.

To focus on the detailed information of each wind farm node, each wind farm is regarded as a node in the graph structure. A mask $\boldsymbol{m}^a \in \boldsymbol{R}^{H \times W}$ for wind farm node *a* is defined: it is defined as 1 if the node of wind farm *a* exists, otherwise 0. By replacing $M_{h,w}^a$ in Formula (6) with $m_{h,w}^a$, the calculation formula for the feature vector $\boldsymbol{g}_a$ of wind farm node *a* is obtained:

$$\boldsymbol{g}_a = \left(\oplus\left\{\boldsymbol{X} \odot \boldsymbol{Z}_{\mathrm{g}}^* \mid m_{h,w}^a = 1\right\}\right) \tag{10}$$

By replacing $\boldsymbol{G}_i$ and $\boldsymbol{G}_j$ in Formula (8) and Formula (9) with $\boldsymbol{g}_a$ and $\boldsymbol{g}_b$, the final output $\boldsymbol{g}_a$ of wind farm node *a* is obtained as follows:

$$\alpha_{a,b} = \frac{\exp\left(\varsigma\left(\boldsymbol{a}^{\mathrm{T}} \oplus \left(\boldsymbol{U}\boldsymbol{g}_a, \boldsymbol{U}\boldsymbol{g}_b\right)\right)\right)}{\sum_{k \in \Lambda(i)} \exp\left(\varsigma\left(\boldsymbol{a}^{\mathrm{T}} \oplus \left(\boldsymbol{U}\boldsymbol{g}_a, \boldsymbol{U}\boldsymbol{g}_b\right)\right)\right)} \tag{11}$$

$$g_a^* = \sigma\left| \sum_{j \in \Lambda(a)} \alpha_{a,b} \boldsymbol{U}\boldsymbol{g}_b \right| \tag{12}$$

The complexity of graph attention is $O(fkc)$, which is related to the number of nodes *f*, the average number of neighbors *k* per node, and the number of node features *c*. In reality, the number of wind farm nodes is much larger than that of cluster nodes. For example, there are 40 wind farm nodes in this paper, but only 3 cluster nodes. Due to the large number of wind farm nodes, if the graph structure is a dense graph and *k* is close to *f*, the complexity is approximately $O(f^2 c)$, resulting in significant computational overhead. Therefore, for the node feature extraction module, it is necessary to increase the sparsity of the graph structure to improve computational efficiency. Thus, this paper proposes a Dynamic Hybrid Sparse Graph Attention (DHS-GAT), and its implementation is as follows:

*a) Prior Node Selection:* Prior Node Selection. Based on the historical power of each wind farm, the correlation coefficient of the wind farm's historical power is calculated. For wind farm *a*, the top $e_1$ wind farms most correlated with it are selected to obtain the prior node set $\boldsymbol{\xi}_a$ of wind farm *a*. $e_1$ denotes the number of prior nodes.

*b) Dynamic Node Selection:* Based on Formula (11), the attention weights between wind farm *a* and other nodes are calculated, and the top $e_2$ wind farms are selected to obtain the dynamic node set $\boldsymbol{\omega}_a$ of wind farm *a*. $e_2$ denotes the number of dynamic nodes.

*c) Generation of Dynamic Sparse Graph:* Take the union of the prior node set and the dynamic node set to generate the adjacent node set $\boldsymbol{\eta}_a = \boldsymbol{\xi}_a \cup \boldsymbol{\omega}_a$. Connect node *a* with the nodes in $\boldsymbol{\eta}_a$; then Formula (11) can be modified as follows:

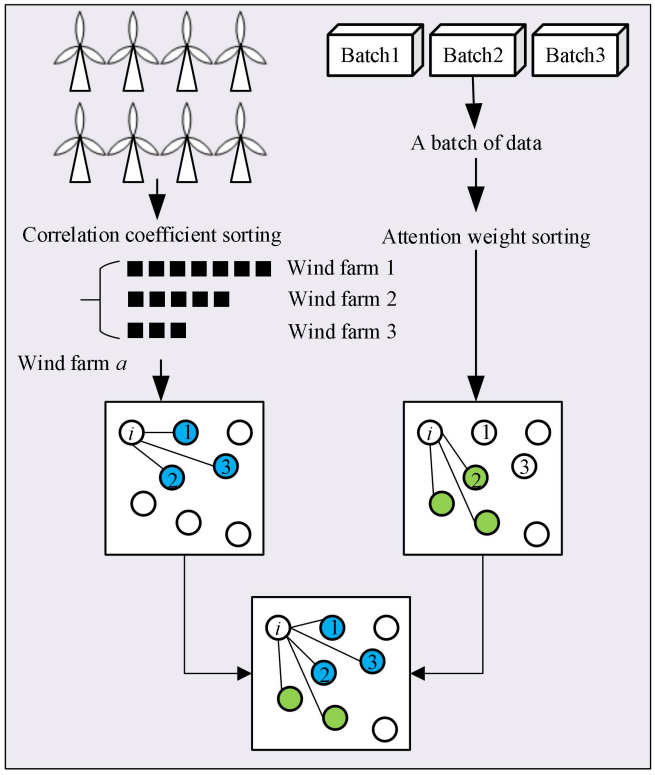


Fig. 5. Generation of dynamic hybrid sparse graphs

$$\alpha_{a,b}=\begin{cases}\dfrac{\exp\left(\varsigma\left(\boldsymbol{a}^{\mathrm{T}}\oplus\left(\boldsymbol{U}\boldsymbol{g}_a,\boldsymbol{U}\boldsymbol{g}_b\right)\right)\right)}{\sum\limits_{k\in\Lambda(a)}\exp\left(\varsigma\left(\boldsymbol{a}^{\mathrm{T}}\oplus\left(\boldsymbol{U}\boldsymbol{g}_a,\boldsymbol{U}\boldsymbol{g}_b\right)\right)\right)}, & b\in\boldsymbol{\eta}_a\\ 0, & b\notin\boldsymbol{\eta}_a\end{cases} \tag{13}$$

The generation method of the dynamic hybrid sparse graph is illustrated in Fig. 5.

## III. Case Study and Analysis

This study conducts case analysis using data from a wind farm cluster in Jilin Province, China, spanning from December 2021 to February 2024. The cluster comprises 40 wind farms with a total installed capacity of 4822 MW. The data from December 2021 to November 2023 is divided into the training set, and the data from December 2023 to February 2024 is divided into the test set. The ratio of training to validation subsets within the training set is 8:2. The data has a time resolution of 1 hour, including grid-based NWP and measured power of each wind farm. The NWP provides the prior-day 2 m pressure, 10 m wind speed, 10 m wind direction, 100 m wind speed, and 100 m wind direction for each spatial node.

The Normal Root Mean Square Error (NRMSE) and Normal Mean Absolute Error (NMAE) are selected as the evaluation metrics for the model. Their calculation formulas are as follows:

$$E_{\mathrm{NRMSE}}=\frac{1}{C}\sqrt{\frac{\sum_{t=1}^{T}\left(R_t-O_t\right)^2}{T}} \tag{14}$$

$$E_{\mathrm{NMAE}}=\frac{\sum_{t=1}^{T}\left|R_t-O_t\right|}{TC} \tag{15}$$

In the formula, $C$ is the installed capacity; $R_t$ is the measured power at time $t$; $O_t$ is the predicted power at time $n$.

The model is built based on Python 3.9 using the PyTorch framework. The hardware and system configurations are as follows: Windows 11 operating system, AMD Ryzen 7 5800H with Radeon Graphics CPU, 16 GB RAM, and Nvidia GeForce RTX 3060 GPU. The hyperparameters of the model are set as follows: learning rate of 0.001, batch size of 32, 50 epochs, data block size of $d_H=d_W=4$, number of prior nodes $\mathrm{A}e_1=4$, and number of dynamic nodes $\mathrm{B}e_2=4$.

### A. Comparison of the Effectiveness of Global Feature Extraction

To verify the effectiveness of the Global Feature Extraction GBA-Transformer (GBA-Tf), four comparison models are used to predict the total power of the cluster based on all NWP data. These models are as follows:

- Convolutional Neural Network-Long Short-term Memory (CNN-LSTM).
- Residual Network (ResNet).
- Transformer (Tf).
- Block-wise Attention Transformer without global variables (BA-Tf).

The prediction performance of each model is shown in Fig. 6. The line "T" in the figure represents the training time of the model, with the unit of seconds.

As shown in Fig. 6, the prediction accuracy of Tf is higher than that of CNN-LSTM and ResNet, but its training time is extremely long, which is unfavorable for model parameter tuning and deployment. By decomposing and combining data, BA-Tf significantly reduces the model complexity, reducing its training time to 6.8% of that of Tf. However, it abandons the global correlation of data during data decomposition, resulting in lower prediction accuracy than Tf. On the basis of BA-Tf, GBA-Tf introduces global variables, enabling the model to achieve a good balance between prediction accuracy and computational efficiency. Not only is its training time much shorter than that of Tf, but compared with Tf, its predicted NRMSE is reduced by 0.465% and NMAE by 0.249%.

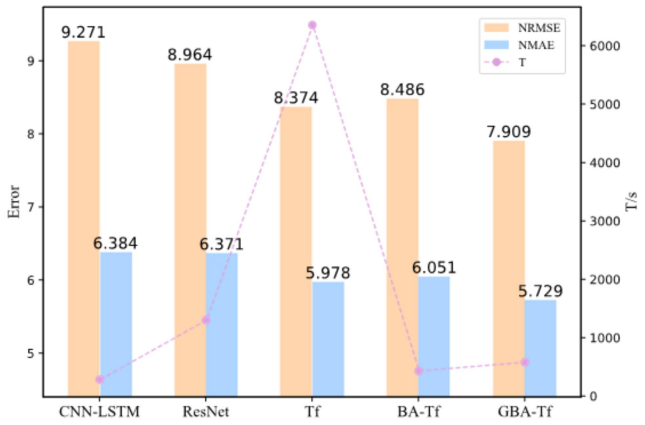


Fig. 6. Predictive performance of global feature extractors

### B. Comparison of the Effectiveness of Node Feature Extraction

To verify the effectiveness of the Node Feature Extraction DHS-GAT, five comparison models are used to predict the total power of the cluster based on the NWP data of wind farm nodes. Among these models:

- C-GAT: A GAT model using a complete graph structure, where all nodes in the graph are directly connected.
- S-GAT: A GAT model using a dynamic sparse graph structure, which does not use prior knowledge; nodes in the graph are dynamically connected based on the ranking of attention scores.

The prediction performance of each model is shown in Fig. 7.

As shown in Fig. 7, due to the introduction of the graph structure, the prediction accuracy of GAT models is higher than that of models without a graph structure. Among the three graph models, DHS-GAT achieves the highest prediction accuracy. Compared with C-GAT, DHS-GAT uses a sparse

connection graph, so its training time is only 50.9% of that of C-GAT, which significantly improves computational efficiency. Compared with S-GAT, DHS-GAT incorporates important node prior knowledge, which can reduce the impact of abnormal data and thus enhance the overall prediction performance.

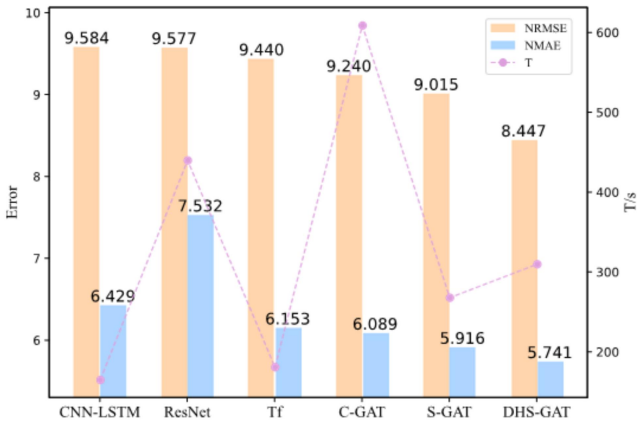


Fig. 7. Predictive performance of node feature extractors

## IV. Conclusions

To address wind power prediction in a wide-area space, this study proposes a short-term power prediction model for wind farm clusters based on features of different spatial scales. The key conclusions are as follows:

For large spatial scales, high-dimensional and complex wide-area meteorological features impose significant computational costs on the self-attention mechanism of the Transformer. Decomposing the overall data into multiple small data blocks can effectively improve the computational efficiency of the self-attention mechanism. Additionally, introducing global variables can mitigate the loss of global information caused by data decomposition.

For the graph attention mechanism, increasing the sparsity of the graph structure can accelerate the model's computation speed. Selecting dynamic nodes based on data structure and prior nodes based on prior knowledge enables the construction of a reasonable graph structure, thereby enhancing the model's generalization ability.

For NWP at large spatial scales, extracting input features separately from the global scale and node scale can improve the model's ability to perceive complex meteorological features, which in turn enhances prediction accuracy.

## References


[1] Niu T, Wang J, Du P. Wpfsad, "Wind power forecasting system integrating dual-stage attention and deep learning," IEEE Transactions on Industrial Informatics, 2023, 19:11252-11264.

[2] Yang B, Zhu T J, Cao P L, et al, "Classification and summarization of solar irradiance and power forecasting methods:a thorough review," Csee Journal of Power and Energy Systems, 2023, 9(3):978-95.

[3] Qiao Kuanlong, Dong Chun, Che Jianfeng, et al, "Short-term prediction method of wind power clusters based on graph convolution neural network under spitio-temporal characteristics," Acta Energiae Solaris Sinica, 2024, 45(5):95-103.

[4] Deng Weisi, Che Jianfeng, Wang Mingqing, et al, "Day-ahead power forecasting method of wind cluster based on grid numerical weather prediction," Southern Power System Technology, 2024, 18(6):51-57+78.

[5] Zhan Weihua, Che Jianfeng, WANG Bo, et al, "A grid-based numerical weather prediction method for multi-output prediction of regional photovoltaic power ," Electric Power, 2024, 18(6):51-57+78.

[6] Nino-adan I, Manjarres D, Landa-torres I, et al, "Feature weighting methods:A review," Expert Systems with Applications, 2021, 184:115424.

[7] Vaswani A, Shazeer N, Parmar N, et al, "Attention is all you need," ArXiv, 2017. [Online]. Available: https://arxiv.org/abs/1706.03762.

[8] Wang S, Li B, Khabsa M, et al, "Linformer: Self-attention with linear complexity," ArXiv, 2020. [Online]. Available: https://arxiv.org/abs/2006.04768.

[9] Nemirko A P, Dula A H, "Machine learning algorithm based on convex hull analysis," Procedia Computer Science, 2021, 186:381-386.

[10] Velickovic P, Cucurull G, Casanova A, et al, "Graph attention networks," ArXiv, 2017. [Online]. Available: https://arxiv.org/abs/1710.10903.

[11] Wang F, Chen P, Zhen Z, et al, "Dynamic spatio-temporal correlation and hierarchical directed graph structure based ultra-short-term wind farm cluster power forecasting method," Applied Energy, 2022, 323:119579.

[12] Shlens J, "A tutorial on principal component analysis," ArXiv, 2014. [Online]. Available: https://arxiv.org/abs/1404.11003.