

Dynamic graph structure and spatio-temporal representations in wind power forecasting

Peng Zang¹, Wenqi Dong¹, Jing Wang¹, and Jianglong Fu^{2,*}

¹ State Grid Jibei Zhangjiakou Wind, PV, Storage and Transmission Renewable Energy Co., Ltd., Zhangjiakou 075000, China

² Hebei University of Architecture, Information Engineering College, Zhangjiakou 075000, Hebei, China

Received: 5 July 2024 / Accepted: 13 November 2024

Abstract. Wind Power Forecasting (WPF) has gained considerable focus as a crucial aspect of the successful integration and operation of wind power. However, due to the stochastic and unstable nature of wind, it poses a real challenge to effectively analyze the correlations among multiple time series data for accurate prediction. In our study, an end-to-end framework called Dynamic Graph structure and Spatio-Temporal representation learning (DSTG) framework is proposed to achieve stable power forecasting by constructing graph data to capture the critical features in the data. Specifically, a Graph Structure Learning (GSL) module is introduced to dynamically construct task-related correlation matrices *via* backpropagation to mitigate the inherent inconsistency and randomness of wind power data. Additionally, a dual-scale temporal graph learning (DTG) module is further proposed to explore the implicit spatio-temporal features at a fine-grained level using different skip connections from the constructed graph data. Finally, comprehensive experiments are performed on the collected Xuji Group Wind Power (XGWP) dataset, and the results show that DSTG outperforms the state-of-the-art spatio-temporal methods by 10.12% on the average of root mean square error and mean absolute error, demonstrating the effectiveness of DSTG. In conclusion, our model provides a promising approach.

Keywords: Graph, Wind power, Spatio-temporal features, Graph neural networks, Dynamic graph structure.

1 Introduction

Recently, with the rapid consumption of fossil fuels, it has become an urgent task to accelerate the development of renewable energy [1]. Wind energy, characterized by its cleanliness, wide distribution and abundant reserves, has become the most promising renewable energy source for large-scale development. With technological advances, wind power is maturing to achieve greater scalability and commercial viability. Actually, wind power is considered as a crucial strategic choice for sustainable energy development with its low emission and non-polluting characteristics [2]. It is worth noting that the intermittent, fluctuating and stochastic nature of wind power poses significant challenges to the safe operation and scheduling of the power grid system when integrating wind turbines into the grid [3, 4]. Therefore, accurately predicting the output power of grid-connected wind farms can significantly mitigate the adverse effects of wind power integration on the power system, promoting the development of scientific power

generation plans, reducing operational costs, and enhancing wind power competitiveness [5].

Specifically, interest in developing WPF methods has grown over the past two decades and has been divided into four categories [6]: (1) Physical-based models [7, 8] rely on numerical weather prediction [9] to determine meteorological information for future time intervals. By analyzing the topography, wake effects and spatial correlations near wind turbines, microscale meteorological data, such as wind speed and direction at turbine hub height, can be obtained. The obtained meteorological data are then mapped onto the power curve of the turbine according to the principle of conservation of mass, momentum and energy to solve the output power of the turbine. However, physical-based methods rely on large amounts of data related to topography and meteorology and require the collaboration of experts from multi-fields to construct the model, which limits their portability and applicability. (2) Statistical models involve attempting to explore the relationship between historical weather conditions and wind power to predict future wind power. Various statistical approaches [10, 11] have been applied in the WPF task, such as autoregressive moving average model [12] and Bayesian network [13].

* Corresponding author: fjl1976@hebiace.edu.cn

Alanazi *et al.* compared different numerical estimation methods for assessing wind energy potential using the Weibull distribution model. Their study shows that this statistical model has high accuracy in wind energy assessment and optimizes the deployment of wind farms [47]. Bedewy *et al.* used GIS spatial analysis to estimate suitable locations for power plant distribution. This approach significantly enhanced the efficiency and scientific rigor of site selection [48]. Statistical methods are generally easier to implement and require minimal feature engineering. They offer economic models that conserve computational resources. Nonetheless, given the dynamic nature of wind, the prediction performances of these shallow learning approaches are affected by the non-grid and stochastic of the wind power data. (3) Additionally, Machine Learning (ML) methods have been widely used to address the non-linear nature of WPF tasks. Models such as Multi-Layer Perception (MLPs) [14], Support Vector Machines (SVMs) [15], Random Forests (RFs) [16] and Extreme Learning Machines (ELMs) [17] have shown promise in WPF tasks. For example, Balal *et al.* employed machine learning models to forecast power generation using historical weather data. Their results showed that these machine learning algorithms effectively improved the prediction accuracy, contributing to better grid optimization and energy management [49]. Unfortunately, ML methods rely heavily on manual feature engineering, requiring significant domain expertise. (4) Recently, Deep Learning (DL) methods [4, 18] have shown superior capability in capturing and uncovering fairly complex intrinsic correlations. For example, Long Short-Term Memory Networks (LSTMs) [19] and Gated Recurrent Units (GRUs) have been developed to address the long-term dependency problem and further to enhance the precision of WPF tasks [20]. Considering the excellent performance of Convolutional Neural Networks (CNNs) [21] and Recurrent Neural Networks (RNNs), some scholars [22, 23] combined these two methods to achieve better prediction performance than using a single spatial or temporal ones. It is worth noting that the DL approach, while achieving high performance, is limited by the need for grid data as input, ignoring the interconnections between the multiple variables that affect the turbine [24].

In fact, the main purpose of the aforementioned four methods is to enhance prediction accuracy by selecting representative local features or phenomena. However, *the complex non-linear functional relationship between wind power and input variables poses a challenge to the approaches that adequately evaluate the implicit importance of each input variable through individual feature extraction and correlation calculations.* Inspired by the success of the Graph Convolution Networks (GCNs) [25, 26] in handling non-grid data, researchers have explored the graph representation method in WPF tasks [27, 28]. In these approaches, each variable is associated with a node in the graph, while connections between multiple-variables represent edges. Notably, using GCN approaches to complete WPF tasks faces two challenges:

Challenge 1: A crucial requirement for GCNs is the construction of graph connectivity (adjacency matrix), which typically involves the use of a pre-determined fixed graph structure *e.g.* the Pearson Correlation Coefficient

(PCC) [29]. However, due to the inherent randomness of wind power data, fixed adjacency matrix construction methods cannot fully exploit the dynamic nature of data. Consequently, determining an appropriate graph structure to ensure precise forecasting becomes a challenging task.

Challenge 2: As shown in Figure 1, the single variable wind speed, for example, exhibits variations at each time of the day, thus wind power forecasting emphasizes a dynamic process, which involves one or more influencing variables. However, the grid DL approaches do not consider the dynamic contexts of wind power, *i.e.*, the stochasticity and correlation of multivariate structural information. To effectively model the spatio-temporal dynamics of wind power, it is crucial to consider both the characteristics of the signal and global structural correlations among multiple variables.

To address these challenges, a solution for WPF is introduced *via* a dynamic spatio-temporal graph representation learning framework, DSTG, which constructs a task-related graph structure and explores stochasticity in multivariate. Specifically, DSTG comprises two components: a graph structure learning (GSL) module and a dual-scale temporal graph learning (DTG) module.

1. GSL employs a supervised scheme to explore important graph adjacency matrices, which dynamically learns the graph structure by using the Ensemble Empirical Mode Decomposition (EEMD) [30] features computed on each time series. This approach allows for adaptive correlation exploration and facilitates the integration of relationships across multiple variables.
2. DTG consists of a graph attention-like cell and a temporal graph-aware LSTM network, which collaboratively models dynamic graph information using spatial feature learning and temporal structure embedding. DTG is further enhanced by a recurrent-skip component that integrates multi-scale temporal features. Furthermore, DTG introduces a temporal smoothness regularization to optimize prediction by capturing fine-grained discriminative features and exploring correlations across temporal graphs.

In conclusion, our contributions are as follows:

- Graph structure learning is designed to dynamically construct the relationship of each element in input, where the relationships (edges) between each variable are fitted by the neural networks. Importantly, the graph structure learning strategy can be guided by the objective function, significantly improving the perceivability of the structure to the task.
- The proposed DTG is a joint spatial-temporal wind prediction module that combines the graph structure learning strategy with a graph attention-like network, and uses a recurrent skip-based LSTM as a temporal feature extractor. It can capture dynamic spatial and multiple temporal features *via* temporal smoothness constraints, thus improving prediction performance.

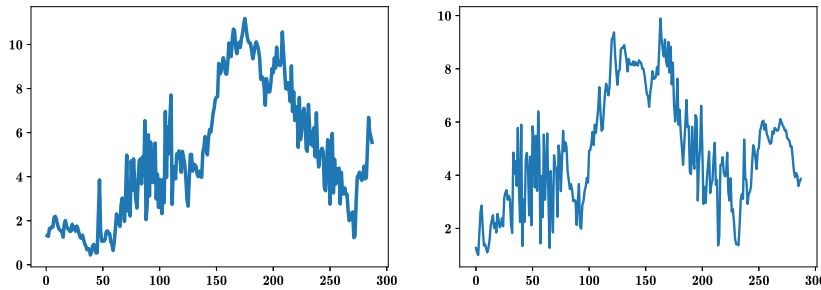


Figure 1. Wind speed intensity for Turbine 1 over two consecutive days. The left graph represents the first two days, while the right graph shows the following two days. The horizontal axis indicates time (in min), and the vertical axis represents wind speed (in m/s).

- A real wind power dataset, the Xuji Group Wind Power (XGWP) dataset, is collected. Extensive experiments conducted on this dataset show that the proposed DSTG is very effective compared to the state-of-the-art methods.

The rest of the paper is organized as follows:

Section 2 represents the related work of this paper. Section 3 details the definition and mathematical formulation of DSTG. Section 4 describes the datasets and presents experiments validating our model’s effectiveness. Finally, conclusions and future work are summarized in Section 5.

2 Related work

Graph convolutional networks (GCNs) and WPF:

Recently, GCNs are particularly suitable for processing non-grid data, including wind power data, due to their ability to model multivariables as nodes in a graph structure, capturing complex and irregular relationships that are not inherently structured in a regular grid format. Based on the manner of neighborhood aggregation, these methods are classified into two categories: 1) *Spatial methods* primarily conduct convolution operations directly on the nodes and their neighboring nodes within the graph. The graph sample and aggregate (GraphSAGE) model [31] performs neighborhood aggregation to generate node embeddings by sampling and aggregating features of neighboring nodes. The Graph Attention Networks (GAT) [32] utilize masked self-attention mechanisms to assign different weights to different nodes in a neighborhood to capture the importance of graph representations. (2) *Spectral methods* define the convolution operation on the graph’s spectral representation. For instance, a generalized graph convolution framework, inspired by the graph Laplacian [33], is proposed to deal with graphs. To reduce the computational cost, Chebyshev’s expansion of the graph Laplacian [34] is utilized, which enhances the efficiency without losing accuracy.

Furthermore, the use of GCNs has become more prevalent in capturing spatial correlations among wind turbine data and exploring the spatial structure through predefined graph structure (*adjacency matrix*) methods, for example: Khodayar and Wang [35] used general LSTMs to extract temporal features and incorporated a spectral

graph convolution to capture spatial features for the WPF task using a mutual information-based adjacency matrix. Li [27] integrated the GCNs and residual network to realize wind power forecast using the PCC-based adjacency matrix. Bentsen *et al.* [36] use graph attention networks for wind power forecasting. Yue Song *et al.* [28] combined maximum information coefficient with GCNs and utilized the multi-resolution convolution neural networks to enhance wind power forecasting by integrating spatial correlations and temporal dynamics. Overall, existing GCN models typically rely on a fixed graph structure, which poses a significant limitation in that the graph structure changes over time in response to external factors, and thus a fixed graph structure cannot be used for accurate prediction.

3 Methods

Our goal is to investigate the interdependencies among multivariate variables, aiming to construct a more precise wind power forecasting framework from a spatio-temporal perspective. Specifically, each influential variable represents a critical external or internal feature, corresponding to a time-series observation. Formally, $\mathbf{X} \in \mathbb{R}^{N \times C}$ is used to indicate the observation data of wind turbines, where N denotes the number of features, and C is the length of the time-series. We use $\mathbf{X}_t = \{x_1, x_2, \dots, x_n\}_{i=1}^N \in \mathbb{R}^{N \times 1}$ to denote the wind power data of the turbine at t timestamp. The WPF task aims to forecast the wind power at future times based on historical observations, *i.e.*, given \mathbf{X} , the objective is to fit a nonlinear function $\phi(\cdot)$ that predicts the wind power supply at future t' steps from the observations at previous t steps as:

$$[\mathbf{X}_{(T-t+1)}, \dots, \mathbf{X}_T] \xrightarrow{\phi(\cdot)} [\hat{y}_{(T+1)}, \dots, \hat{y}_{(T+t')}] \quad (1)$$

where $\hat{Y} \in \mathbb{R}^{t' \times 1}$ represent the predicted active power.¹ This section first summarizes the proposed framework. Then, the specifics of each module are presented, explaining its components and functions.

¹ The key symbols and their corresponding definitions are shown in the [Appendix](#).

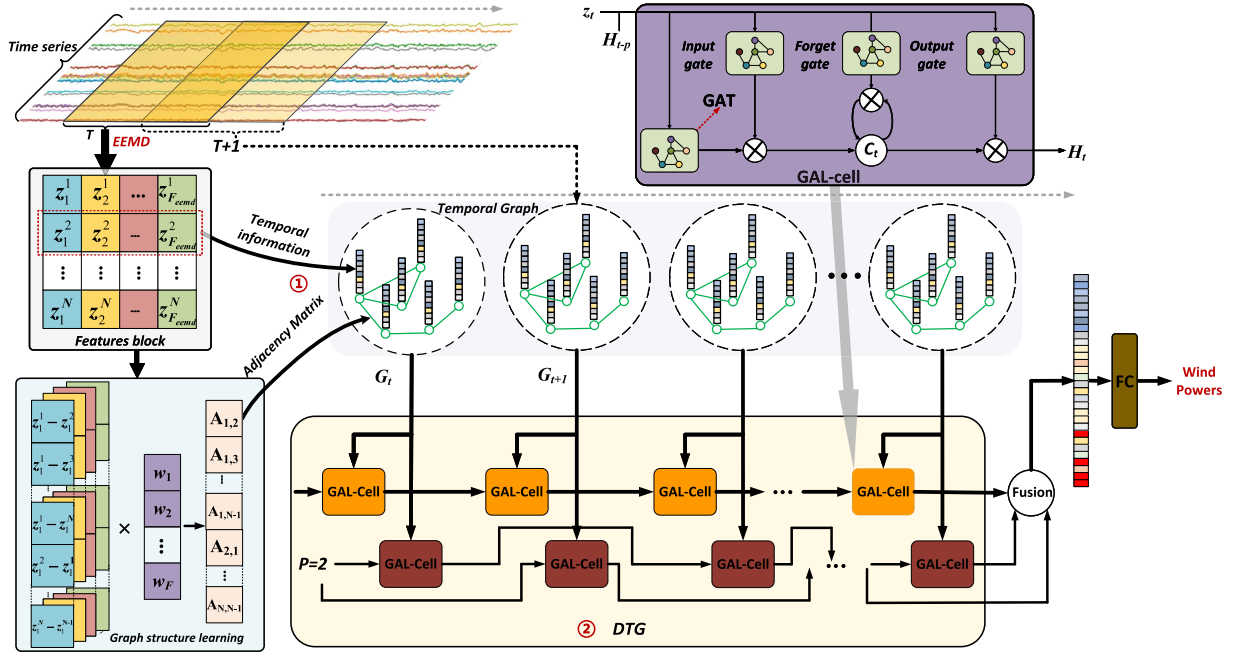


Figure 2. The general framework of DSTG consists of two phases: ① The adaptive graph structure learning module is trained to acquire the dynamic spatial graph structure using the extracted features. ② Then, the dual scale temporal graph learning module is proposed to focus on the extraction of discriminative features by both the graph attention network and LSTM network with the recurrent-skips.

3.1 Overview of the proposed DSTG framework

This study presents a Dynamic Spatio-Temporal Graph (DSTG) representation learning framework, illustrated in Figure 2, which comprises two key phases: (i) During the temporal graph construction phase, the EEMD is applied to extract representative features from the segment data. Subsequently, the graph structure learning module is utilized to process the multivariate input and dynamically construct the correlation adjacent matrix. The extracted features and adjacent matrices are combined to construct the temporal graph series. (ii) In the prediction phase, the dual-scale temporal graph learning module is employed to explore implicit relationships and temporal features from the graph. Here, a recurrent-skip mechanism is employed to facilitate long-term wind power prediction.

3.2 Graph Structure Learning (GSL) module

First, the spatial structural information among the multivariate time series is exploited. Formally, the temporal graphs (undirected) are defined as $G_T = (V, \mathcal{E}, \mathbf{A})$, where V indicates the set of nodes and each one represents a signal time series; $V \in \mathbb{R}^{1 \times N}$, and N indicates the number of multivariates. \mathcal{E} denotes the set of edges, representing the connections between two nodes, and \mathbf{A} denotes the adjacency matrix of G_T . It is worth emphasizing that the structure \mathbf{A} used in our proposed model is obtained through an end-to-end learning process. As shown in Figure 2-①, the raw time series sequences is defined as

$\mathbf{X} = (x_1, x_2, \dots, x_N) \in \mathbb{R}^{N \times C_i}$, where C_i indicates the time series length of i -th variable $x_i \in \mathbf{X} (i \in \{1, 2, \dots, N\})$. For each feature x_i , the differential ensemble empirical mode decomposition (EEMD) [37] is utilized to extract features from various time series. Subsequently, the feature matrix $\mathbf{F}_i = (\mathbf{z}_1^i, \mathbf{z}_2^i, \dots, \mathbf{z}_N^i)^T \in \mathbb{R}^{N \times F_{\text{emmd}}}$ is defined for each series x_i , where $\mathbf{z}_n^i \in \mathbb{R}^{F_{\text{emmd}}}$ ($n \in \{1, 2, \dots, N\}$) denotes the extracted \mathbf{F}_{emmd} features of node n at series i . Moreover, the graph structure is learned instead of relying on a graph constructed based on prior knowledge or artificial. Specifically, an adaptive function $\mathbf{A}_{mn} = g(\mathbf{z}_m, \mathbf{z}_n)$ is defined to represent the connection relationship between variable \mathbf{z}_m and \mathbf{z}_n based on the input feature matrix $\mathbf{F}_i = (\mathbf{z}_1^i, \mathbf{z}_2^i, \dots, \mathbf{z}_N^i)^T \in \mathbb{R}^{N \times F_{\text{emmd}}}$, where $m, n \in \{1, 2, \dots, N\}$. $g(\mathbf{z}_m, \mathbf{z}_n)$ is implemented through a Fully-Connected (FC) layer with the trainable weight $\mathbf{w} = (w_1, w_2, \dots, w_{F_{\text{emmd}}})^T \in \mathbb{R}^{F_{\text{emmd}} \times 1}$. Each edge \mathcal{E}_{mn} of the learned graph structure (adjacency matrix) \mathbf{A} is defined as follows:

$$\mathcal{E}_{mn} = g(\mathbf{x}_m, \mathbf{x}_n) = \frac{\exp(\sigma(\mathbf{w}^T |\mathbf{z}_m - \mathbf{z}_n|))}{\sum_{n=1}^N \exp(\sigma(\mathbf{w}^T |\mathbf{z}_m - \mathbf{z}_n|))}, \quad (2)$$

where $\sigma(\cdot)$ is a leaky rectified linear unit function to ensure that each edge \mathcal{E}_{mn} in the adjacency matrix \mathbf{A} is non-negative. Moreover, the softmax function [50] is used to normalize each row of \mathbf{A} . Particularly, the weight matrix \mathbf{w} is learned and updated by minimizing the following loss function,

$$\mathbf{L}_{\text{gsl}} = \sum_{m,n=1}^N \|\mathbf{z}_m - \mathbf{z}_n\|_2^2 \cdot \mathcal{E}_{mn} + \lambda \|\mathbf{A}\|_F^2. \quad (3)$$

It means that the larger distance $\|\mathbf{z}_m - \mathbf{z}_n\|_2$ between \mathbf{z}_m and \mathbf{z}_n , the smaller the corresponding value of \mathcal{E}_{mn} . To guarantee the sparsity of the graph \mathbf{A} in the correlation connection structure, the ℓ_2 regularization term [51] is employed, where λ is a constraint term.

3.3 Dual scale Temporal Graph (DTG) learning module

To capture the dynamic spatio-temporal patterns in temporal graphs, the DTG module is introduced to comprehensively learn the temporal characteristics present in time-series data from two distinct aspects: multivariable spatial features and temporal graph information.

As shown in Figure 2-②. (i) To fully explore the spatio-temporal dynamics of wind power data, EEMD features from the time series of each node are used as their corresponding initial embeddings to build temporal graphs. Graph attention-like networks are then utilized to capture spatial features at the graph level; however, these networks are incapable of capturing temporal characteristics. In contrast, adaptive graph attention networks are integrated into the LSTM architecture (referred to as GAL) to effectively exploit spatio-temporal features in modeling dynamic wind power networks. (ii) The traditional LSTM is limited by long-term dependency and cannot exploit dependencies with variable lengths. To overcome this limitation, dynamic skip connections are proposed to extend the temporal span and enrich the optimization information, specifically by adding dynamic jump connections between the current hidden cell and the neighboring hidden cells.

Formally, GAL takes temporal graph features with node embeddings as inputs and consists of three gate components: the input gate \mathcal{I} , the forget gate \mathcal{F} and the output gate \mathcal{O} . Particularly, each gate operation is replaced with a stack of adaptive graph attention layers (GAL-cell) to capture spatial features of the temporal graphs. The input of the GAL-cell consists of three components: \mathcal{H}_{t-p} , \mathbf{A}_t , and \mathbf{E}_t^G . Here, \mathcal{H}_{t-p} represents the hidden state obtained in the $(t-p)$ time-step, where p denotes the number of hidden cells skipped. The matrix \mathbf{A}_t represents the adjacency matrix that is learned for the t -th graph structure within the input temporal graph series. It is worth noting that our graph attention is constructed differently from traditional GAT [32], and we learn a new way of constructing node attention weight relies on GSL. On the other hand, \mathbf{E}_t^G refers to the feature matrices associated with the t -th graph structure. Unlike traditional LSTM [43] that operates solely on temporal embedding, our approach utilizes both graph structure and temporal features as inputs. The updating procedure of gates is expressed as follows:²

1) input gate \mathcal{I}_t ,

$$\mathcal{I}_t = \sigma(\mathbf{W}_{x_i} \cdot f_{\text{gat}}(\hat{G}_t) + \mathbf{W}_{h_i} \cdot f_{\text{gat}}(H_{t-p}) + B_i), \quad (4)$$

2) forget gate \mathcal{F}_t ,

$$\mathcal{F}_t = \sigma(\mathbf{W}_{x_f} \cdot f_{\text{gat}}(\hat{G}_t) + \mathbf{W}_{h_f} \cdot f_{\text{gat}}(H_{t-p}) + B_f), \quad (5)$$

3) output gate \mathcal{O}_t ,

$$\mathcal{O}_t = \sigma(\mathbf{W}_{x_o} \cdot f_{\text{gat}}(\hat{G}_t) + \mathbf{W}_{h_o} \cdot f_{\text{gat}}(H_{t-p}) + B_o), \quad (6)$$

4) input modulation gate \mathcal{U}_t ,

$$\mathcal{U}_t = \sigma(\mathbf{W}_{x_c} \cdot (\hat{G}_t) + \mathbf{W}_{h_c} \cdot f_{\text{gat}}(H_{t-p}) + B_c), \quad (7)$$

5) cell state update \mathcal{C}_t ,

$$\mathcal{C}_t = \psi(\mathcal{I}_t \cdot \mathcal{U}_t + \mathcal{F}_t \cdot \mathcal{C}_{t-p}), \quad (8)$$

6) hidden state \mathcal{H}_t ,

$$\mathcal{H}_t = \mathcal{O}_t \cdot \psi(\mathcal{C}_t), \quad (9)$$

where $f_{\text{gat}}(\cdot)$ indicates the graph convolution operation and B is the bias term, $\psi(\cdot)$ is the Tanh activate function, and the \mathbf{E}_t^G is graph embedding of \hat{G}_t . The GAT convolution operation for l -th layer can be written as

$$\mathbf{E}_t^{G^{(l+1)}} = \sigma(\mathbf{A}_t \cdot \mathbf{W}^l \cdot \mathbf{E}_t^G), \quad (10)$$

where matrix \mathbf{W}^l is the trainable weight of l -th layer, $\mathbf{E}_t^{G^l}$ and $\mathbf{E}_t^{G^{(l+1)}}$ are the node embedding computed before and after the GAT operation at l -th and $(l+1)$ -th steps, respectively. In particular, $\mathbf{E}_t^{G^{(0)}}$ is the graph embedding of \hat{G}_t at the first layer, which is equal to \mathbf{E}_t^S , and each item \mathbf{A}_t can be obtained by equation (2). Then, the outputs of the GAL-cell are combined to produce a final embedding as follows:

$$\hat{\mathcal{H}}_t^C = \mathbf{W}^R \cdot \mathcal{H}_t^R + \sum_{j=1}^p \sum_{i=1}^j \mathbf{W}_i^S \cdot \mathcal{H}_{t-i}^S + B_{sk}, \quad (11)$$

where \mathcal{H}_t^R is the hidden state of recurrent-skip component at time t , $\{\mathcal{H}_{t-p+1}^S, \mathcal{H}_{t-p+2}^S, \dots, \mathcal{H}_t^S\}$ are the p hidden states from time stamp $t-p+1$ to t , matrices \mathbf{W}^R and \mathbf{W}_i^S are the trainable weights. Finally, $\hat{\mathcal{H}}_t^C$ is flattened into a vector along the node dimension and fed into FC layers for prediction.

3.3.1 Learning and optimization

The loss function uses the mean squared error,

$$\mathbf{L}_{\text{mse}} = \frac{1}{\tau} \sum_{i=1}^{\tau} (y_i - \hat{y}_i)^2, \quad (12)$$

where, y_i denotes the true wind power values, \hat{y}_i are the predicted wind power, and τ indicates the total number of time points. Moreover, a regularization constraint, temporal smoothness, is proposed to capture the temporal correlations in the progression of wind power. Our assumption is that there is a minimal difference between power values at two consecutive time points. We model the temporal smoothness by penalizing significant differences between the predicted values at consecutive time points as follows:

$$\mathbf{L}_s = \sum_{i>2}^{\tau} (\hat{y}_i - \hat{y}_{i-1})^2. \quad (13)$$

² All of these formulas build on LSTMs [43].

Hence, the total loss of DSTG is as follows:

$$\mathbf{L}_{\text{total}} = \mathbf{L}_{\text{mse}} + \gamma \mathbf{L}_{\text{gsl}} + \lambda \mathbf{L}_{\text{s}}, \quad (14)$$

where γ and λ are constant values of the constraint term. With supervision, the DSTG model can be optimized using back-propagation to achieve a better solution. Importantly, to reduce the time-intensive task of identifying an optimal solution, our method incorporates a multi-time window ensemble strategy. We train multiple DSTG instances using varying window lengths on the provided temporal graph series. This approach enables the models to capture distinct temporal patterns by considering various time strides. Finally, the predictions of these trained models are then combined using a majority voting mechanism to yield a final prediction.

4 Experiments an results

The experiments are designed to explore and answer the following questions, which will help in validating the effectiveness of the proposed modules:

- Q1.** Is DSTG more effective than the state-of-the-art method?
- Q2.** Is GSL effective for constructing correlations between multiple variables?
- Q3.** What is the contribution of each component to the DTG module?

4.1 Datasets and experimental setup

Datasets: The goal of our work is to predict the current wind turbine power using the weather conditions detected by each wind turbine and the internal state of the turbines. Accordingly, a dataset, XGWP, was collected from the Zhangjiakou wind farm energy storage company operated by China Xuji Group. The dataset includes data from 8 doubly-fed turbines, each rated at 2.0 MW. The data are mainly divided into two primary parts: historical meteorological data, turbine data and historical power data. 1) The first part is the numerical weather data, collected from the wind farm’s anemometer tower and calibrated with historical weather forecast data. 2) The second part comprises turbine status data and power output data obtained from the wind farm’s Supervisory Control And Data Acquisition (SCADA) system. The dataset spans the period from July 4, 2020, to December 31, 2020, and includes data from each turbine sampled at 10-minute intervals, yielding 25,920 valid records. For more details on the XGWP dataset, refer to Table 1. For a fair comparison, the same preprocessing procedures and operations as outlined in the code³ are followed to analyze zero values, missing data, unknown data, and to perform anomaly detection. Additionally, XGWP is sequentially split into train, validation, and test datasets in chronological order by 150 days, 20 days, and 10 days, respectively, to predict future active power time series of length 288.

³ https://github.com/PaddlePaddle/PaddleSpatial/tree/main/apps/wpf_baseline_gru.

Experiment settings: DSTG is implemented using PyTorch and the Adam optimizer with a decay rate of 1×10^{-4} to mitigate over-fitting, with the training process consisting of 100 epochs. To ensure model stability, the initial learning rate is set to 1×10^{-4} and decayed gradually with the StepLR strategy. Each hidden layer utilized a dropout rate of 0.2. To smooth the historical power data and focus on data trends, the sliding window is applied with lengths of 576, 432, 288 and a stride of 10, respectively.

4.2 Evaluation metrics

To evaluate the prediction of DSTG, the final evaluation score s_t^i for wind turbine i at time step t is calculated as the average of the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) of the prediction results,

$$s_t^i = \frac{1}{2} \left(\sqrt{\frac{\sum_{j=1}^{\tau} (y_{t+j}^i - \hat{y}_{t+j}^i)^2}{\tau}} \right) + \frac{1}{2} \left(\frac{\sum_{j=1}^{\tau} |y_{t+j}^i - \hat{y}_{t+j}^i|}{\tau} \right), \quad (15)$$

where y and \hat{y} are the true and predicted wind power, respectively, and τ is the total prediction time-step number. The total score S_t is reported, *i.e.*, the total prediction scores for all wind turbine scores s_t^i . For testing, roll over the test dataset using the prediction window and report the average evaluation score. Additionally, the Mean Absolute Percentage Error (MAPE) [44] is reported to offer an intuitive measure of the prediction’s deviation from the actual power value.

4.3 Comparison with state-of-the-art methods (Q1)

To demonstrate the validity of DSTG, three categories of methods are compared using original settings from their respective papers. To maintain the fairness of the experimental results and ensure stability in validation, five identical experiments (with different random seeds) are performed on the XGWP dataset, and the evaluation indicator S_t is averaged to obtain the final results. Here, this paper uses the paired t -test at a 5% significance level to assess statistically significant differences between DSTG and other methods.

DSTG is compared with the following methods, as shown in Table 1, all of which use features such as Etmp, Wspd and Wdir as inputs.

GRU is a sequential prediction model that utilizes a single gating unit to concurrently regulate the forgetting factor and determine whether to update the state unit.

AutoFormer [25] is a decomposition architecture that performs dependency discovery and information aggregation at the sequence level by embedding sequence decomposition blocks as internal operators.

SCINet [42] combines downsampling, convolution and interaction in a hierarchical structure, employing varied convolution filters and iterative operations for handling information across multiple temporal scales.

GWNET [41] is a model that combines the concepts of GCNs and wavelet transforms to learn features in graph-structured data at multiple scales.

Table 1. Column names and their definitions.

Column	Column name	Specification
1	TurbID	Wind turbine Identifier
2	Day	Day of the record
3	Tmstamp	Recorded time
4	Wspd (m/s)	Wind speed recorded by the anemometer
5	Wspd (m/s)	Filtered wind speed
6	Wdir ($^{\circ}$)	Angle of wind direction and nacelle position
7	Etmp ($^{\circ}C$)	Temperature of the surrounding environment
8	Itmp ($^{\circ}C$)	Temperature in the nacelle of the turbine
9	Tspd (rpm)	Turbine speed
10	Ndir ($^{\circ}$)	Deflection angle of the nacelle
11	Pab1 ($^{\circ}$)	Blade one: Pitch angle
12	Pab2 ($^{\circ}$)	Blade two: Pitch angle
13	Pab3 ($^{\circ}$)	Blade three: Pitch angle
14	Gens (rpm)	Generator Speed
15	Reac (kW)	Reactive power
16	Actp (kW)	Active power (real power value)

Ast-GCN [26] combines the spatio-temporal GCNs to capture spatial interactions and to construct features at each important time step.

RBCNN [21] is a two-step hybrid strategy, initially extracting features through variable mode decomposition, followed by transforming these features into images. It then applies modified residual-based CNNs for WPF.

DCRNN [38] is a data-driven model that combines diffusion convolutional layers and RNN layers to capture spatial dependencies and temporal dynamics in time series, enabling accurate predictions of future data.

AGCRN [39] combines node adaptive parameter learning and data-adaptive graph generation, enhancing GCNs for spatial correlation discovery and node pattern learning.

HSTTN [40] enhances WPF by integrating hierarchical temporal modeling, transformer-based spatio-temporal dependencies and contextual fusion blocks.

GCMCN [28] combines GCNs with multi-resolution CNNs, and enhances wind power forecasting by integrating spatial correlations and temporal dynamics.

Mic-LSTM [45] enhances WPF by denoising SCADA data with wavelet decomposition and selecting key features using the maximum information coefficient to train a LSTM network.

Mvmd-DNN [46] enhances WPF by integrating multivariate variational mode decomposition, elastic net variable selection and a hybrid neural network (CNN, BiLSTM and Attention) to capture the coupling between wind power series and multiple meteorological series.

Based on the results in Table 2, DSTG exhibits significant differences from other methods in both MAE and RMSE metrics, with p -values less than 0.05, indicating that the predictive performance of DSTG is statistically significantly better than the other methods. Moreover, Figure 3 shows the comparison between the predicted power curve

and the actual power curve of our DSTG model, where the dashed and solid lines indicate the actual and predicted power curves, respectively. Particularly, the MAPE metrics between the predicted power and the true power for each method are shown in Table 2. The following conclusions can be drawn:

1. The RNN-based approaches outperform the CNN-based approaches by 2.68 scores, which is attributed to the fact that CNNs primarily focus on extracting local feature patterns and overlook the spatial characteristics of wind turbines. Additionally, the GCN-based method obtains lower MAPE scores and performs better in capturing spatial information, while the traditional CNN-based method is difficult to effectively capture the spatial structure in multivariate data. In particular, transformer-based spatio-temporal methods are less effective in wind power forecasting when the data are limited and lack periodicity. Similarly, RNN-based models are not well-suited for aggregating spatial features from all wind turbine data.
2. Compared with the single GCN-based approaches (*e.g.*, GCMCN), our spatio-temporal GCN method attempts to exploit temporal information by learning and explicitly adjusting spatial information. It can fully exploit the spatio-temporal correlation of the data while alleviating the uncertainty caused by random variables, thus achieving better performance.
3. Importantly, our graph-based methods leverage spatial structure learning to alleviate the data dynamic problem with a stable optimization process. Figure 4 illustrates the optimization process of DSTG. It can be observed that constructing graph structures at a finer level facilitates the stable learning of relationships between multivariate, thus obtaining better performance.

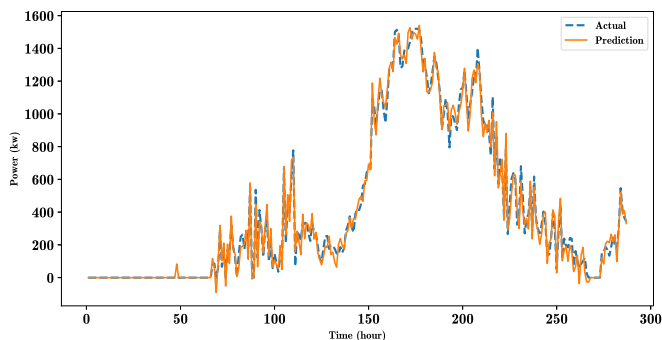
Table 2. Results comparisons between DSTG and other methods, where ST indicates spatio-temporal.

Type	Methods	Scores	MAE	RMSE	MAPE
S/T	GRU ^[tw]	42.4213 [†]	37.2985	47.5441 [†]	14.21%
	AutoFormer ^[tw]	48.4166 [†]	42.8654 [†]	53.9678 [†]	27.40%
	SCINet ^[tw]	42.1947 [†]	38.0369 [†]	46.3524	17.68%
	Mic-LSTM ^[tw]	43.4459 [†]	39.2348 [†]	47.6571	13.64%
	Mvmd-DNN ^[tw]	41.9165 [†]	37.5376 [†]	46.2957	9.45%
S/T -CNN	GWNET ^[tw]	49.3725 [†]	43.1823 [†]	55.5627 [†]	14.12%
	Ast-GCN ^[tw]	44.3207 [†]	39.8564 [†]	48.7850 [†]	10.35%
	RBCNN ^[tw]	48.6482 [†]	43.2697 [†]	54.0266 [†]	13.22%
	HSTTN ^[tw]	43.7319 [†]	38.3510	49.1128 [†]	11.76%
	S/T -RNN	DCRNN ^[tw]	44.1880 [†]	39.8136 [†]	48.5623 [†]
AGCRN ^[tw]		46.7399 [†]	41.0236 [†]	52.4561 [†]	10.22%
GCMCN ^[tw]		43.3202 [†]	39.7753 [†]	46.8650	10.63%
DSTG ^[tw]		41.1007	36.6652	45.5362	8.40%

* T represents the temporal aspect, while S denotes the spatial aspect.

** The best metrics are bolded and those with [†] Indicate significant differences.

*** Methods use external temperature, wind speed or both as inputs are labeled [t], [w] or [tw], respectively.

**Figure 3.** Diagram of real power and predicted power.

4.4 Ablation studies

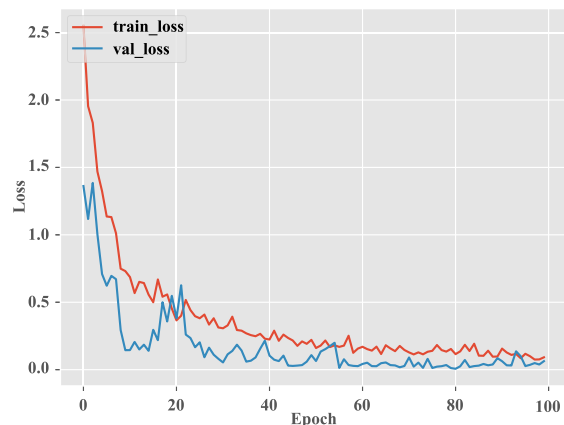
In this section, the effects of model hyperparameters in DSTG will be systematically explored. Specifically, the impacts of the following factors are discussed:

- (i) the effectiveness of each proposed component.
- (ii) the impact of recurrent-skip number.
- (iii) the impact of node information aggregation manners.

4.4.1 Effectiveness of each proposed component (Q2)

The proposed DSTG framework comprises three key components: the graph structure learning (GSL) module, the dual scale temporal graph learning (DTG) module and the recurrent-skips (RS). To assess the impact of each component within DSTG, several variants have been designed for comparative analysis:

1. **DSTG -w/o GSL** a variant of DSTG with graph structure learning is removed from the framework, replacing it with a PCC-based adjacency matrix.

**Figure 4.** Loss curves for the optimization process, including training and validation.

2. **DSTG -w/o DTG** a variant of DSTG, where no DTG module and all graph embeddings are fed directly into the fully-connected (FC) layer for prediction.
3. **DSTG -w/o RS** a variant of DSTG, where no recurrent-skips setting and all graph embeddings are fed directly into the LSTM layer for prediction.

Specifically, the DSTG model is compared with three of its variants. From the results shown in Table 3, our work finds that 1) each component has different contributions in different parts. The performance decreases after removing each component, demonstrating the complementary and necessity of these three components. 2) Comparing DSTG -w/o GSL with DSTG reveals that the joint learning of graph structure is beneficial to explore the shared

Table 3. Ablation studies of the components in DSTG.

Method	Scores	MAE	RMSE
DSTG <i>w/o</i> GSL	43.6999	39.0643	48.3354
DSTG <i>w/o</i> DTG	47.6157	44.8853	50.3461
DSTG <i>w/o</i> RS	44.2983	42.3552	49.2414
DSTG	41.0959	36.8651	45.3267

correlation among the multivariate, thereby extracting relevant dynamic embeddings from latent variate attributes and separating specific spatial information. 3) By comparing DSTG *w/o* DTG with DSTG, it can be verified that the association and fusion of graph convolution and LSTM significantly affect the performance of prediction. 4) DSTG *w/o* RS significantly reduces the prediction performance of the model, suggesting that our recurrent-skips setup aggregates the information more efficiently and better captures the long-distance-dependent features for the WPF task.

4.4.2 Impact of different recurrent-skip length (Q3)

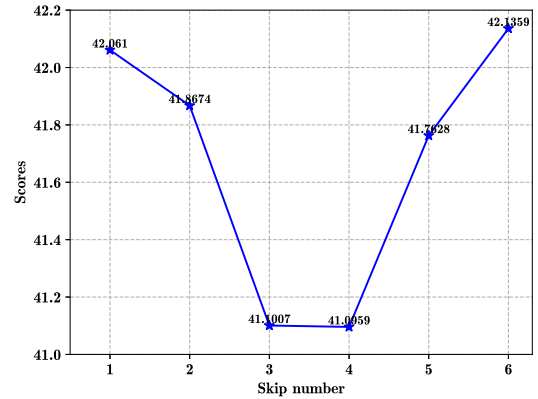
To examine the impact of the skip-connection component proposed in our work, an ablation study with varying skip lengths is conducted. The results of the experiments are presented in Figure 5.

It is evident that the performance of the model improves with an appropriate skip length, indicating the effectiveness of the skip-connection component. The incorporation of the recurrent-skip component significantly enhances the model's performance, demonstrating its ability to capture both long-term and short-term temporal patterns in wind power data. Nevertheless, it is worth emphasizing that an excessively large number of connections may lead to information redundancy and potentially degrade the model's performance. Hence, it is crucial to strike a balance and select an optimal skip length to achieve the best results.

4.4.3 Impact of node aggregation manners (Q3)

To assess the effectiveness of the GAL-cell design, a comprehensive ablation study is performed comparing DSTG with two simpler models: GCN+LSTM and GraphSAGE+LSTM. Specifically, temporal graph embeddings are acquired via GCN and GraphSAGE networks, which are subsequently fed into LSTM modules.

As depicted in Table 4, the results highlight the effectiveness of GAL-cell, validating our motivation to incorporate graph convolution into LSTM while considering the dynamic variation of the graph structure. This approach proves capable of modeling spatio-temporal dependencies in wind power data. Moreover, the ordinary GCN and GraphSAGE models ignore dynamic correlations among multivariate variables and do not effectively preserve the correlations that do exist. In contrast, GAL-cell adaptively retains correlations among multivariables, which helps alleviate the issues associated with the inherently stochastic nature of the wind power data.

**Figure 5.** Ablation study on recurrent-skips.**Table 4.** Ablation study on combination different GCNs with LSTM.

Method	Score	MAE	RMSE
DSTG <i>w/</i> GCN	43.0959	38.8651	47.3267
DSTG <i>w/</i> GAT	41.1007	36.6652	45.5362
DSTG <i>w/</i> GraphSAGE	43.8794	39.6206	48.1382

4.5 Limitations of our work

The findings presented in this work consistently demonstrate the efficacy of the DSTG model. Nevertheless, there are three drawbacks: first, the dataset used is small and may not be able to fully optimize the parameters of the GSL module. It is suggested that future work could improve the performance of DSTG by incorporating a pre-training phase on a wider range of datasets. Second, the DTG module employs several recurrent-skips to capture variations in the different time-scales, which is effective but also increases computational cost. Additionally, it is crucial to consider factors such as the number of parameters and inference speeds related to real-time predictions in practical applications.

5 Conclusion and future work

This study introduces the Dynamic Spatio-Temporal Graph (DSTG) model, a novel framework designed for wind power forecasting that effectively captures spatio-temporal correlations and addresses the stochastic nature of wind power data. Our approach leverages dynamic graph embeddings to adapt to spatial-temporal dynamics, significantly enhancing prediction accuracy. The key components of our approach include: 1) Adaptive graph structure learning: This component dynamically integrates spatial information into the graph structure, continuously adjusting to changing data dynamics, which is crucial for accurate forecasting in volatile environments like wind power generation. 2) Dual-scale temporal graph learning: By capturing both micro and macro temporal dynamics, this module

ensures a comprehensive understanding of time-dependent patterns, improving the model's ability to forecast over varying time horizons. Our extensive experiments on the XGWP dataset show that DSTG performs significantly better than existing state-of-the-art methods. Specifically, DSTG achieved a 10.12% improvement in the average of root mean square error and mean absolute error compared to the best-performing existing methods. Notably, the flexibility of DSTG allows it to be applied not only to the dataset used in this study, but also suggests its potential for wider application in renewable energy forecasting. By directly addressing the core challenges in wind power forecasting, our model opens up new avenues for research in energy prediction, particularly in harnessing the power of graph-based learning mechanisms in other complex, dynamic systems. Future work will focus on exploring the integration of additional predictive variables to enhance its forecasting capabilities. Furthermore, practical considerations for implementing the DSTG framework in real-time wind power forecasting will be prioritized, particularly addressing latency and efficiency challenges. This includes optimizing model response times, implementing model compression techniques, and ensuring the framework can adapt to varying operational conditions to maintain robust performance in real-time scenarios.

Funding

This research is funded by the Science and Technology Project of State Grid Hebei Zhangjiakou Wind, Storage and Transmission New Energy Co.-(GWZJK2019).

Conflicts of interest

All authors declare that they have no competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Deng X., Shao H., Hu C., Jiang D., Jiang Y. (2020) Wind power forecasting methods based on deep learning: A survey, *Comput. Model. Eng. Sci.* **122**, 1, 273–302.
- Foley A.M., Leahy P.G., Marvuglia A., McKeogh E.J. (2012) Current methods and advances in forecasting of wind power generation, *Renew. Energy* **37**, 1, 1–8.
- Wang X., Guo P., Huang X. (2011) A review of wind power forecasting models, *Energy Proc.* **12**, 770–778.
- Wang H.Z., Li G.Q., Wang G.B., Peng J.C., Jiang H., Liu Y. T. (2017) Deep learning based ensemble approach for probabilistic wind power forecasting, *Appl. Energy* **188**, 56–70.
- Tastu J., Pinson P., Trombe P.J., Madsen H. (2013) Probabilistic forecasts of wind power generation accounting for geographically dispersed information, *IEEE Trans. Smart Grid* **5**, 1, 480–489.
- Chen C., Liu H. (2020) Medium-term wind power forecasting based on multi-resolution multi-learner ensemble and adaptive model selection, *Energy Convers. Manag.* **206**, 112492.
- Landberg L. (1998) A mathematical look at a physical power prediction model, *Wind Energy* **1**, 1, 23–28.
- Chen P., Pedersen T., Bak-Jensen B., Chen Z. (2009) ARIMA-based time series model of stochastic wind power generation, *IEEE Trans. Power Syst.* **25**, 2, 667–676.
- Wang Q., Liu W., Yu H., Zheng S., Gao S., Granelli F. (2019) CPAC: Energy-efficient algorithm for IoT sensor networks based on enhanced hybrid intelligent swarm, *Comput. Model. Eng. Sci.* **121**, 1, 83–103.
- Milligan M., Schwartz M., Wan Y.H. (2003) *Statistical wind power forecasting models: Results for US wind farms*, NREL/CP-500-33956, National Renewable Energy Lab (NREL), Golden, CO (United States).
- Sideratos G., Hatzigiorgiou N.D. (2007) An advanced statistical method for wind power forecasting, *IEEE Trans. Power Syst.* **22**, 1, 258–265.
- Yuan X., Tan Q., Lei X., Yuan Y., Wu X. (2017) Wind power prediction using hybrid autoregressive fractionally integrated moving average and least square support vector machine, *Energy* **129**, 122–137.
- Adedipe T., Shafiee M., Zio E. (2020) Bayesian network modelling for the wind energy industry: An overview, *Reliab. Eng. Syst. Saf.* **202**, 107053.
- Sun W., Wang Y. (2018) Short-term wind speed forecasting based on fast ensemble empirical mode decomposition, phase space reconstruction, sample entropy and improved back-propagation neural network, *Energy Convers. Manag.* **157**, 1–12.
- Hu Q., Zhang S., Yu M., Xie Z. (2015) Short-term wind speed or power forecasting with heteroscedastic support vector regression, *IEEE Trans. Sustain. Energy* **7**, 1, 241–249.
- Zeng J., Qiao W. 2011 Support vector machine-based short-term wind power forecasting, in: *2011 IEEE/PES Power Syst. Conf. Expos.*, IEEE, pp. 1–8.
- Liu H., Tian H.Q., Li Y.F. (2015) Four wind speed multi-step forecasting models using extreme learning machines and signal decomposing algorithms, *Energy Convers. Manag.* **100**, 16–22.
- Hong Y.Y., Rioflorida C.L.P.P. (2019) A hybrid deep learning-based neural network for 24-h ahead wind power forecasting, *Appl. Energy* **250**, 530–539.
- Liu X., Zhou J. (2024) Short-term wind power forecasting based on multivariate/multi-step LSTM with temporal feature attention mechanism, *Appl. Soft Comput.* **150**, 111050.
- Chen J., Zeng G.Q., Zhou W., Du W., Lu K.D. (2018) Wind speed forecasting using nonlinear-learning ensemble of deep learning time series prediction and extremal optimization, *Energy Convers. Manag.* **165**, 681–695.
- Yildiz C., Acikgoz H., Korkmaz D., Budak U. (2021) An improved residual-based convolutional neural network for very short-term wind power forecasting, *Energy Convers. Manag.* **228**, 113731.
- Yin H., Ou Z., Huang S., Meng A. (2019) A cascaded deep learning wind power prediction approach based on a two-layer of mode decomposition, *Energy* **189**, 116316.
- Liu H., Mi X., Li Y., Duan Z., Xu Y. (2019) Smart wind speed deep learning based multi-step forecasting model using singular spectrum analysis, convolutional Gated Recurrent Unit network and Support Vector Regression, *Renew. Energy* **143**, 842–854.
- Zhao Z., Yun S., Jia L., Guo J., Meng Y., He N., Yang L. (2023) Hybrid VMD-CNN-GRU-based model for short-term forecasting of wind power considering spatio-temporal features, *Eng. Appl. Artif. Intell.* **121**, 105982.
- Wu H., Xu J., Wang J., Long M. (2021) Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, *Adv. Neural Inf. Process. Syst.* **34**, 22419–22430.

- 26 Zhou H., Ren D., Xia H., Fan M., Yang X., Huang H. (2021) Ast-gnn: An attention-based spatio-temporal graph neural network for interaction-aware pedestrian trajectory prediction, *Neurocomputing* **445**, 298–308.
- 27 Li H. (2022) Short-term wind power prediction via spatial temporal analysis and deep residual networks, *Front. Energy Res.* **10**, 920407.
- 28 Song Y., Tang D., Yu J., Yu Z., Li X. (2022) Short-term forecasting based on graph convolution networks and multiresolution convolution neural networks for wind power, *IEEE Trans. Ind. Inf.* **19**, 2, 1691–1702.
- 29 Liu Z., Li M., Yang B., Wei H. (2022) *Spatial wind power forecasting using a GRU-based model*. Baidu KDD Cup, WindTeam CSU123.
- 30 Wang S., Zhang N., Wu L., Wang Y. (2016) Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method, *Renew. Energy* **94**, 629–636.
- 31 Hamilton W., Ying Z., Leskovec J. (2017) Inductive representation learning on large graphs, *Adv. Neural Inf. Process. Syst.* **30**, 1025–1035.
- 32 Veličković P., Cucurull G., Casanova A., Romero A., Lio P., Bengio Y. (2017) Graph attention networks, *Stat* **1050**, 20, 10–48550. arXiv preprint arXiv:1710.10903.
- 33 Bruna J., Zaremba W., Szlam A., LeCun Y. (2013) *Spectral networks and locally connected networks on graphs*, arXiv preprint arXiv:1312.6203.
- 34 Defferrard M., Bresson X., Vandergheynst P. (2016) Convolutional neural networks on graphs with fast localized spectral filtering, *Adv. Neural Inf. Process. Syst.* **29**, 3844–3852.
- 35 Khodayar M., Wang J. (2018) Spatio-temporal graph deep neural network for short-term wind speed forecasting, *IEEE Trans. Sustain. Energy* **10**, 2, 670–681.
- 36 Bentsen L.Ø., Warakagoda N.D., Stenbro R., Engelstad P. (2022) May) Wind park power prediction: Attention-based graph networks and deep learning to capture wake losses, *In J. Phys.: Conf. Ser* **2265**, 2, 022035. IOP Publishing.
- 37 Wu Z., Huang N.E. (2009) Ensemble empirical mode decomposition: a noise-assisted data analysis method, *Adv. Adapt. Data Anal.* **1**, 01, 1–41.
- 38 Li Y., Yu R., Shahabi C., Liu Y. (2018) Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, *ICLR-2018*. arXiv preprint arXiv:1707.01926.
- 39 Bai L., Yao L., Li C., Wang X., Wang C. (2020) Adaptive graph convolutional recurrent network for traffic forecasting, *Adv. Neural Inf. Process. Syst.* **33**, 17804–17815.
- 40 Zhang Y., Liu L., Xiong X., Li G., Wang G., Lin L. (2023) Long-term wind power forecasting with hierarchical spatial-temporal transformer, *IJCAI-2023*, 6308–6316. arXiv preprint arXiv:2305.18724.
- 41 Xu B., Shen H., Cao Q., Qiu Y., Cheng X. (2019) Graph wavelet neural network, *ICLR-2019*. arXiv preprint arXiv:1904.07785.
- 42 Liu M., Zeng A., Chen M., Xu Z., Lai Q., Ma L., Xu Q. (2022) Scinet: Time series modeling and forecasting with sample convolution and interaction, *Adv. Neural Inf. Process. Syst.* **35**, 5816–5828.
- 43 Yu Y., Si X., Hu C., Zhang J. (2019) A review of recurrent neural networks: LSTM cells and network architectures, *Neural Comput.* **31**, 7, 1235–1270.
- 44 De Myttenaere A., Golden B., Le Grand B., Rossi F. (2016) Mean absolute percentage error for regression models, *Neurocomputing* **192**, 38–48.
- 45 Liu Z.H., Wang C.T., Wei H.L., Zeng B., Li M., Song X.P. (2024) A wavelet-LSTM model for short-term wind power forecasting using wind farm SCADA data, *Expert Syst. with Appl.* **247**, 123237.
- 46 Yang T., Yang Z., Li F., Wang H. (2024) A short-term wind power forecasting method based on multivariate signal decomposition and variable selection, *Appl. Energy* **360**, 122759.
- 47 Alanazi M.A., Aloraini M., Islam M., Alyahya S., Khan S. (2023) Wind energy assessment using Weibull distribution with different numerical estimation methods: a case study, *Eme. Sci. J.* **7**, 6, 2260–2278.
- 48 Bedewy B.A.H., Al-Timimy S.R.A. (2023) Estimate suitable location of solar power plants distribution by GIS spatial analysis, *Civil Eng. J.* **9**, 5, 1217–1229.
- 49 Balal A.T., Jafarabadi Y.P.T., Demir A.T., Igene M.T., Giesselmann M.T., Bayne S.T. (2023) Forecasting solar power generation utilizing machine learning models in Lubbock, *Eme. Sci. J.* **7**, 4, 1052–1062.
- 50 Sharma S., Sharma S., Athaiya A. (2017) Activation functions in neural networks, *Towards Data Sci.* **6**, 12, 310–316.
- 51 Van Laarhoven T. (2017) *L2 regularization versus batch and weight normalization*, arXiv preprint arXiv:1706.05350.

Appendix

A.1 Major symbols and definitions

As shown in Table A1, a list of key symbols along with their corresponding definitions is provided.

Table A1. The major symbols and their definitions.

Symbol	Definition
$\mathbf{X} \in \mathbb{R}^{N \times C}$	Observation data matrix of wind turbines with N features and C time points.
$\mathbf{X}_t \in \mathbb{R}^{N \times 1}$	Wind power data vector at timestamp t , and x_i represents the i -th feature.
$\phi(\cdot)$	Nonlinear function for DSTG model fitting.
$\hat{Y} \in \mathbb{R}^{t' \times 1}$	Predicted active power values for t' time steps.
$G_T = (V, \mathcal{E}, \mathbf{A})$	Temporal graph, where V is the set of nodes, \mathcal{E} is the edge set, and \mathbf{A} is the adjacency matrix.
$V \in \mathbb{R}^{1 \times N}$	Set of nodes representing multivariate signals.
$\mathbf{F}_i \in \mathbb{R}^{N \times F_{\text{emmd}}}$	Feature matrix for each time series variable x_i after EEMD.
\mathcal{E}_{mn}	Edge weight representing the connection between nodes \mathbf{z}_m and \mathbf{z}_n .
$\mathbf{A}_{mn} = g(\mathbf{z}_m, \mathbf{z}_n)$	Adaptive function to determine edge connectivity based on node features.
\mathbf{L}_{gsl}	Loss function for graph structure learning incorporating distance and regularization.
$\mathcal{I}, \mathcal{F}, \mathcal{O}$ and \mathcal{H}	Gates in the GAL-cell: input, forget, output gates and hidden states.
w	Trainable weight vector for GSL module.
\mathbf{W}	Trainable matrices for different DTG layers.
\mathbf{E}_t^G	Feature matrices for the t -th graph structure.
\mathbf{L}_{mse}	Mean squared error loss between true and predicted values.
\mathbf{L}_s	Regularization terms for time smoothness.
$\mathbf{L}_{\text{total}}$	Total loss function for model optimization.