Received: 19 May 2023 | Revised: 20 July 2023 | Accepted: 13 August 2023

DOI: 10.1049/rpg2.12825

IET Renewable Power Generation

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Diffusion-based conditional wind power forecasting via channel attention

Hongqiao Peng[1] | **Hui Sun**[1] | **Shuxin Luo**[1] | **Zhengmin Zuo**[1] | **Shixu Zhang**[2] | **Zhixian Wang**[3] | **Yi Wang**[3]

[1]Planning Research Center of Guangdong Power Grid Corporation CSG, Guangzhou, Guangdong Province, China

[2]Tsinghua Sichuan Energy Internet Research Institute, Building 5, Area A, Tianfu Elite Center, Science City, Chengdu, Sichuan, China

[3]Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong Special Administrative Region of China, China

Correspondence
Zhixian Wang, Department of Electrical and Electronic Engineering, The University of Hong Kong Hong Kong Special Administrative Region of China.
Email: zxwang@eee.hku.hk

Funding information
Power Planning Special Topic of Guangdong Power Grid Co., Ltd., titled "Research on Multi-time-space Scale Renewable Energy Output Characteristics Simulation and Typical Mode Identification Methods in Guangdong Province", Grant/Award Number: 031000QQ00220002

Abstract

Wind energy is one of the most significant renewable sources of energy while accurate and reliable wind power forecasting methods may greatly benefit power system planning and scheduling. Recently, many machine learning algorithms have shown significant advantages in how to extract temporal features for wind power forecasting. However, wind power curves in the time domain frequently display intermittent features and significant uncertainty, which is not favorable to precise and reliable forecasting. In this paper, the Diffusion and Channel Attention-based Wind Power Forecasting (DC-WPF) framework is proposed, which transforms wind power data into the frequency domain while applying advanced channel attention techniques to aid the model in capturing the frequency domain information and ultimately enhancing accuracy. With high-accuracy results, DC-WPF then proposes a diffusion-based framework to transform the point forecasting results into probabilistic forecasts to capture the uncertainty. Finally, extensive experiments on six wind power plants show that our method can improve the point forecasting accuracy of wind power and provide advanced probabilistic forecasts at a multi-time scale.

1 | INTRODUCTION

1.1 | Backgrounds and motivations

In the modern world, renewable energy sources like wind and photovoltaic (PV) are increasingly replacing conventional fossil fuels [1]. This is due to the energy system's reliance on conventional fossil fuels, which are responsible for more than 75% of the world's carbon dioxide (CO2) emissions and serious environmental impact [2]. Countries all over the world have established necessary legislation to encourage the growth of renewable energy to decrease potential risks to the environment from the energy sector. Among all renewable energies, Wind power generation is a significant one and is used extensively in many nations throughout the globe. CAs of the end of 2017, China had 635 million kW installed capacity for the generation of renewable energy, which made up 35.7% of the country's installed electric power capacity. Additionally, 164 million kW of installed wind power, or 9.2% of total installed capacity, is available [3]. The European Council has repeatedly set objectives for the proportion of renewable energy in recent years, and in many EU member states, wind energy is predicted to make up the bulk of these goals [4].

However, because of the variability of wind [5, 6] and the fact that the wind power sources are exposed to the environment more than traditional thermal power sources [7], the uncertainty in wind power generation is very large, which probably leads to detrimental impacts on the electric grid. To solve such challenges, researchers are focusing heavily on wind power forecasting, and power system operators may reduce the risk of an unstable energy supply with precise and reliable wind power predictions, which also allows them to

make better decisions concerning the expansion of the wind sector [8].

Even though accuracy and reliability are both important for wind power forecasting. We seldom consider the relationship between them. Therefore, this paper has two main concerns. The first is to improve wind power point forecasting accuracy. The other is to provide advanced probabilistic predictions based on point forecasting and illustrate how point forecasting influences the probabilistic one to capture uncertainty, which may ultimately increase wind power forecasting reliability.

1.2 | Literature review

Extensive work has been done for wind power forecasting, which can be roughly divided into three categories: persistence method, statistical methods, and methods based on neural networks. The persistence method refers to simply considering the current value as the future value based on the assumption that they are highly correlated [9]. This kind of assumption is unrealistic that it ignores the volatility of wind power. Therefore, it only performs well in the very short-term situation. To overcome such difficulties, some researchers applied statistical methods like AR models and their variants ARX, ARMA, and ARIMA to model the wind power data [10–12]. This model assumes that the wind power sequence is auto-correlated and tries to find the exact correlation. However, AR models only focus on the temporal features of time series and have few parameters, making it difficult to accurately model complex wind power generation.

With the development of computing power these years, the neural network has become a powerful tool for wind power forecasting. It is believed that neural networks can fit arbitrary functions with enough parameters [13]. In terms of high-accuracy wind power forecasting, one of the networks called long short-term memory(LSTM) has been proven to be very suitable for wind power forecasting tasks with a large number of experiments [14–16]. In addition, [17] proposed an architecture of MST-GNN based on wind Transformer and GNN to give advanced wind power forecasts. [18] proposed the Transformer model, which is widely used in natural language processing, to replace temporal context with an attention mechanism. In addition to pursuing high accuracy, many researchers also use neural networks to provide probabilistic forecasting results to improve the reliability of forecasting. [19] used an ensemble method combining several distributions to give probabilistic results. [20] calculated the point-based wind power prediction's uncertainty and obtained the wind power's probabilistic prediction interval using particle swarm optimization (PSO).

Even though neural networks have strong modeling capabilities, wind power exhibits strong volatility in the time domain, making it difficult to model effectively. Therefore, a natural idea is to introduce frequency domain information to avoid such difficulties [21]. Some researchers have used methods such as Fourier Transform(FT) to transform wind power sequences into the frequency domain and extract features [22] and [23] proposes a novel integrated method for short-term wind power forecasting, which first exact the frequency domain information by fast Fourier transform(FFT). However, Fourier Transform itself has certain limitations. According to [24], Fourier Transform will introduce high-frequency components incorrectly because of its problematic periodicity. And this will cause an error value for boundary information, which is called the Gibbs phenomenon. As shown in Figure 1, most of the wind power energy is contained in the rather low-frequency range. Incorrect introduction of high-dimensional components can make it difficult for the model to extract information correctly. Therefore, how to correctly extract frequency domain information to improve the accuracy and reliability of forecasting deserves our attention.

To cope with such a situation, [25] adopts wavelet transform to transform the wind power temporal information into the frequency domain and input them into the convolutional neural network to extract features. [26] proposes tests various combinations of Recurrent Kalman Filter (RKF), Fourier Series (FS), Wavelet (WNN), and Artificial Neural Network (ANN) to finally get 12 different hybrid models for forecasting. Unlike simply using neural networks to extract frequency domain information, combining frequency domain information with attention mechanisms has become a new trend in time series forecasting. [27] proposes a sequence-to-sequence (Seq2Seq) LSTM model with attention mechanisms and wavelet transform for reservoir-level forecasting. Based on wavelet transform (WT), [28] decomposes and reconstructs the time series of crude oil futures prices into a low-frequency main sequence and several high-frequency noise sequences and uses the BiLSTM-Attention-CNN model to forecast the decomposition subsequences in turn. [29] proposes a frequency-enhanced decomposition structure to decompose the time series and integrate the Fourier enhancement module and wavelet enhancement module into the encoder and decoder of the Transformer. However, these forecasting models are not specific to wind power forecasting, and they all focus on point forecasting. How to combine the extraction of wind power sequence frequency domain information with an attention mechanism to provide more accurate forecasting results still needs to be studied. Furthermore, how to extend the point forecasting model to probabilistic forecasting to provide reliable forecasting also needs to be considered.

1.3 | Contributions

To address the existing research gaps mentioned above and provide accurate and reliable wind power forecasting, this paper proposes a framework called DC-WPF. The framework combines a frequency domain-based attention mechanism enhancement method with a conditional diffusion model to improve the accuracy of wind power deterministic forecasting while providing corresponding probabilistic forecasting results, which enhances the reliability of the forecasting model.

The contributions of this paper are summarized below:

- To solve the problem that complex temporal features of wind power make it difficult for the model to exact useful things for the following forecasting, we combine a

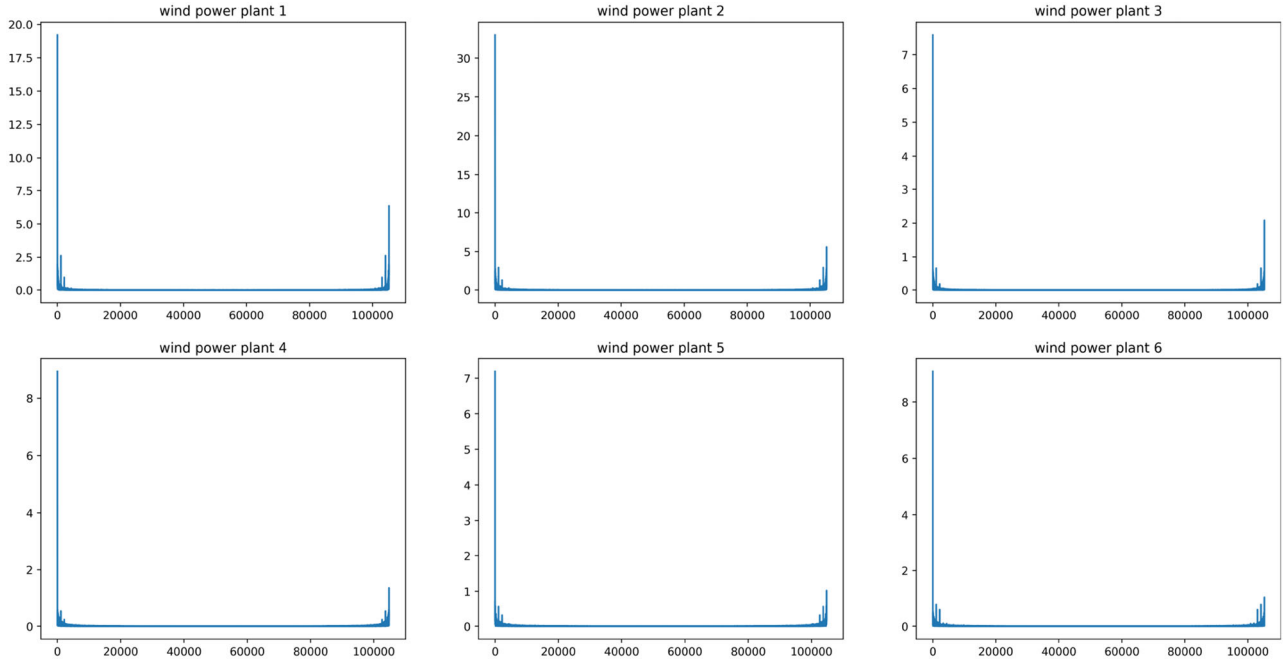

FIGURE 1 Six real wind power datasets visualization in frequency domain, most of the energy is contained in the low-frequency range.

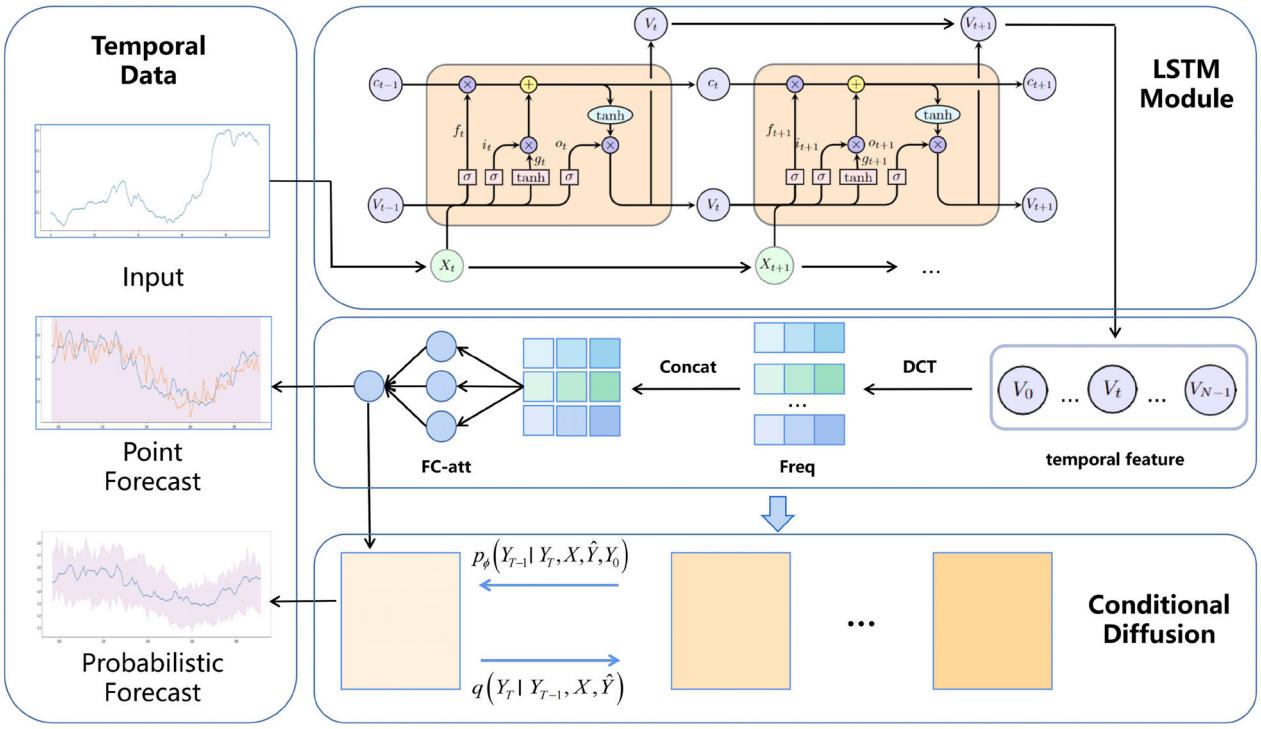

FIGURE 2 The architecture of the proposed framework DC-WPF. The DC-WPF consists of two parts: The upside part is the deterministic forecasting model and the downside part (Conditional Diffusion) is the diffusion transformation module.

frequency-based channel attention mechanism with the LSTM model to enhance the model, thereby greatly improving the accuracy.

- Based on highly accurate deterministic forecasting, we apply conditional diffusion models to expand deterministic forecasting to probabilistic one, enabling it to give forecasting results under arbitrary confidence intervals, effectively estimating the uncertainty of the forecasting model, and improving forecasting reliability.
- Conduct a comprehensive case study based on real wind power datasets at different time scales. The experiments show that considering frequency information has a positive

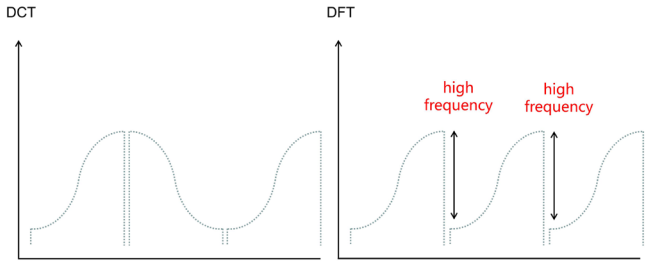

FIGURE 3 Difference of discrete Fourier transform and discrete cosine transform [24].

ALGORITHM 1 Training process of diffusion-based transformation

Data: Deterministic forecasting network P_θ; Diffusion reverse network Q_ϕ; Wind power history data X, label Y_0

while *not convergence* **do**

  for *batch in batch loader* **do**

    $\hat{Y} \leftarrow P_\theta(X)$;

    Take the gradient step on $\nabla_\theta \left\| Y_0 - \hat{Y} \right\|^2$

  Sample $t \sim U(\{1, 2, 3 \dots T\})$;

  Sample $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$;

  Calculate the loss $\mathcal{L}_\epsilon =$ $\left\| \epsilon - \epsilon_\phi(X, \sqrt{\bar{\alpha}_t} Y_0 + \sqrt{1-\bar{\alpha}_t}\epsilon + \left(1 - \sqrt{\bar{\alpha}_t}\right) P_\theta(X), P_\theta(X), t) \right\|^2$

  Take the gradient step on $\nabla_\phi \mathcal{L}_\epsilon$

Result: Deterministic forecasting network P_θ; Diffusion reverse network Q_ϕ

impact on the model at a multi-time scale and our method is competitive when it is compared with current widely used methods.

2 | PROBLEM STATEMENT

We will divide our task into two stages. One is the point forecasting task, which aims to provide high-precision forecastings, while the other is probabilistic forecasting, which aims to provide reliable forecastings.

For point forecasting, our known quantities are the current time point t_0 and historical wind power data before that time point $X_{t_0-b_1:t_0} = \{x_{t_0-b_1}, x_{t_0-b_1+1}, \dots, x_{t_0}\}$ and the quantity we hope to obtain is the predicted value of wind power output in the future $X_{t_0+1:t_0+b_2} = \{x_{t_0+1}, x_{t_0+2}, \dots, x_{t_0+b_2}\}$,

$$X_{t_0+1:t_0+b_2} = P_\theta(X_{t_0-b_1:t_0}),$$

where b_1 and b_2 represent the input length and the prediction length respectively. By adjusting their length, we can achieve forecastings from a few hours to a day in advance.

Unlike general forecasting scenarios, our probabilistic forecasting is closely related to the results of point forecasting. That is, we need to use the output of point forecasting as the input for probabilistic forecasting, and ultimately obtain the final probabilistic output.

ALGORITHM 2 Inference process of diffusion-based transformation

Data: Deterministic forecasting network P_θ; Diffusion reverse network Q_ϕ; Wind power history data X.

$\hat{Y} \leftarrow P_\theta(X)$;

Sample $Y_T \sim \mathcal{N}\left(\hat{Y}, I\right)$;

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

  Draw $z \sim \mathcal{N}(\mathbf{0}, I)$ if $t > 1$;

  Calculate re-parameterized term $\hat{Y}_0 =$ $\frac{1}{\sqrt{\bar{\alpha}_t}}\left(Y_t - \left(1 - \sqrt{\bar{\alpha}_t}\right)\hat{Y} - \sqrt{1-\bar{\alpha}_t}\epsilon_\phi\left(X, Y_t, \hat{Y}, t, Q_\phi\right)\right)$;

  Let $Y_{t-1} = \gamma_0 \hat{Y}_0 + \gamma_1 Y_t + \gamma_2 \hat{Y} + \sqrt{\tilde{\beta}_t} z$ if $t > 1$, else set $Y_{t-1} = \hat{Y}_0$

Result: Deterministic forecasting result $\hat{Y}$; Probabilistic forecasting result Y_0

$$\mathbf{X_{t_0+1:t_0+h_2}} = \mathcal{Q}_{\boldsymbol{\phi}}(X_{t_0+1:t_0+b_2}),$$

where $\mathbf{X_{t_0+1:t_0+h_2}}$ represents multiple probabilistic sampling samples at each forecasting time point. Through these samples, we can obtain the corresponding probability distribution properties

3 | METHODOLOGY

In this section, we will introduce how the channel mechanism technique can assist our model to incorporate frequency domain information to increase forecasting accuracy and how the diffusion model transforms the deterministic forecasts into probabilistic ones.

3.1 | Overview of the DC-WPF framework

The entire design of our suggested approach DC-WPF is depicted in Figure 2. To extract temporal features, we first feed the wind power data into the LSTM module [30]. Then we use Discrete Cosine Transform (DCT) [31] to transform the wind power data into the frequency domain to avoid the error value for boundary information stated in the introduction. Unlike most frequency-based methods, we do not apply the inverse transform, which can minimize additional computation. Similar to [24], we apply a frequency-enhanced channel attention mechanism to extract frequency features and aggregate the results for the final output. With this output, we build a conditional denoising diffusion model to give the probabilistic results. In the following sections, we will introduce the details of DCT transformation and conditional diffusion models.

3.2 | DCT and channel attention mechanism

3.2.1 | Discrete cosine transform (DCT)

To capture the frequency features of the wind power data, we focus on 1D DCT in this paper. Given a discrete sequence

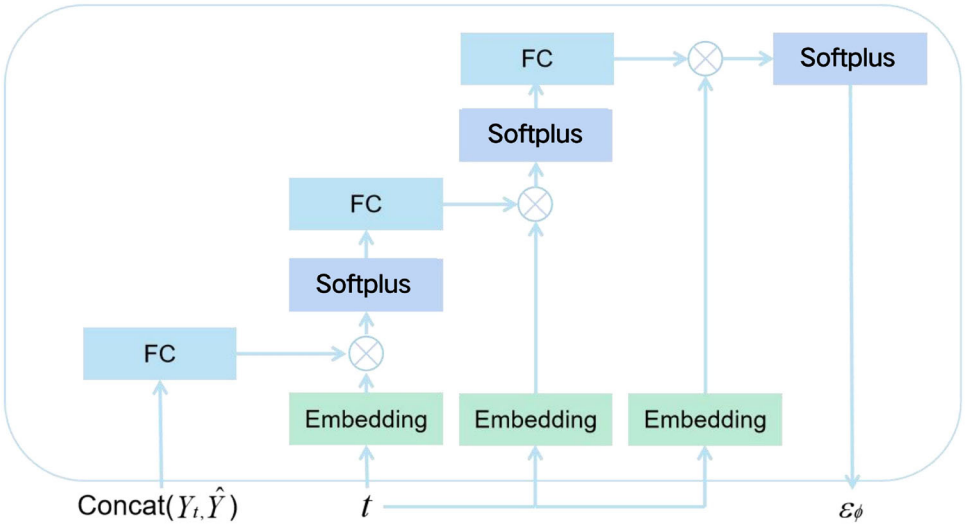

FIGURE 4 The network architecture of the conditional guided model.

$X[k]_{k=0:N-1}$, we can rewrite it as linear combination of multiple components as below,

$$X[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x'[n] \cos\left(\frac{\left(n+\frac{1}{2}\right)\pi k}{N}\right), \tag{1}$$

where $x'[n]$ can be calculated as

$$x'[n] = \sum_{n=0}^{N-1} X[k] cos\left(\frac{\left(n+\frac{1}{2}\right)\pi k}{N}\right). \tag{2}$$

In this way, we generate a frequency domain spectrum from the discrete sequence $X[k]_{k=0:N-1}$. Note that the DCT is similar to DFT (discrete Fourier transform), and the difference between them is that DCT converts the original signal into a real even signal before transformation, avoiding the erroneous introduction of high-frequency components at the periodic edge (shown in Figure 3).

3.2.2 | Channel attention mechanism

As shown in Figure 2, we first input the wind power data into the LSTM module to exact the time domain feature map and we spilt them into N sub-groups(corresponding to C channels). Then we use the DCT transform module to get the frequency spectrum like this

$$\text{Freq}_i^k = DCT_k(V_i) = \sqrt{\frac{2}{L}} \sum_{n=0}^{L-1} V_i'[n] \cos\left(\frac{\left(n+\frac{1}{2}\right)\pi k}{L}\right). \tag{3}$$

After the DCT transformation, we stack all the channels, that is

$$\text{Freq} = DCT(V) = \text{concat}\left(\left[\text{Freq}_0, \text{Freq}_1, \ldots, \text{Freq}_{N-1}\right]\right). \tag{4}$$

Now we get the frequency information and we can use neural network structure like SE block [32] to learn the attention weights like this

$$F_c - att = \sigma(W_2 \delta(W_1 DCT(V))). \tag{5}$$

In this way, each channel features interact with every frequency components and it will encourage networks to enhance the diversity of extracted features [24].

3.3 | Diffusion-based conditional transformation

Inspired by [33], we propose a conditional diffusion transformation method, which is modified to time series data, to transform the deterministic wind power forecasting into probabilistic ones.

The diffusion model already can generate probabilistic results based on the learned distribution. Given the ground-truth Y_0 and its corresponding history data X, we assume that $Y_0 \sim p(Y_0)$. The goal of the denoising diffusion model is to learn the conditional distribution $p_\phi(Y_0|X)$. To achieve this goal, the diffusion model will generate several samples $Y_{1:T}$ concerning T adding noise steps instead of directly generating the Y_0. As a result of the adding noise, we assume the final distribution $p(Y_T|X) = \mathcal{N}(0, I)$ so that we can learn a model through maximizing the log-likelihood $\log p_\phi(Y_0|X)$ and this can be achieved by minimizing the following ELBO derived from variational inference:

TABLE 1 Hyperparameters of the LSTM module.

Hyperparameters	hidden_size	hidden_layers	bias	dropout_rate
	(64,64)	2	True	0(0.1,0.25)

$$\log p_{\phi}(Y_0 \mid X) = \log \int p(Y_T \mid X)\Pi_{t=T}^{1} p_{\phi}(Y_{t-1} \mid X, Y_t) dY_{1:T}$$
$$\geq \underbrace{\mathbb{E}_{q(Y_{1:T}|Y_0,X)}\left[\log p_{\phi}(Y_{0:T} \mid X) - \log q(Y_{1:T} \mid Y_0, X)\right]}_{\mathcal{L}_{\mathrm{ELBO}}}. \tag{6}$$

Here the $q(Y_{1:T} \mid Y_0, X)$ is the forward process of a Markov chain defined by DDPM [34]. Now since we have T generated data, we can use the Bayes rule to break the original ELBO into T parts as below

$$\mathcal{L}_{\mathrm{ELBO}}(Y_0, X) = \mathcal{L}_0(Y_0, X) + \sum_{t=2}^{T} \mathcal{L}_{t-1}(Y_0, X) + \mathcal{L}_T(Y_0, X), \tag{7}$$

where

$$\mathcal{L}_0(Y_0, X) = \mathbb{E}_q[-\log p_{\phi}(Y_0 \mid Y_1, X)], \tag{8}$$

$$\mathcal{L}_{t-1}(Y_0, X) \tag{9}$$
$$= \mathbb{E}_q[D_{\mathrm{KL}}(q(Y_{t-1} \mid Y_t, Y_0, X) \| p_{\phi}(Y_{t-1} \mid Y_t, X))],$$
$$= \|\tilde{\boldsymbol{\mu}}_t(Y_t, Y_0, P_{\theta}(X)) - \boldsymbol{\mu}_{\phi}(Y_t, Y_0, P_{\theta}(X), t)\|^2, \tag{10}$$

$$\mathcal{L}_T(Y_0, X) = \mathbb{E}_q[D_{\mathrm{KL}}(q(Y_T \mid Y_0, X) \| p(Y_T \mid X))]. \tag{11}$$

Note that in the T step, the $\mathcal{L}_T(Y_0, X)$ is independent of the parameter ϕ. According to [35], $\mathcal{L}_0(Y_0, X)$ has little impact on the result that we can also ignore it. In this way, the diffusion model can learn the distribution $p_{\phi}(Y_0 \mid X)$.

Since we already have advanced deterministic results, we propose to build a conditional diffusion model based on such results like [33]. Unlike the DDPM, we assume that the final step of the diffusion process is to be a normal distribution $\mathcal{N}(P_{\theta}(X), I)$. In this way, we can consider the deterministic results $P_{\theta}(X)$ as the prior knowledge to the relationship between the history data X and the desired distribution $p(Y_0 \mid X)$. To achieve this goal, we need to modify the strategy of adding noise.

With the same setting of the schedule parameters $\{\alpha_i, \bar{\alpha}_i\}_{i=1:T} \in (0, 1)$ and $\{\beta_i, \tilde{\beta}_i\}_{i=1:T} \in (0, 1)$ in [34] and [36], the conditional adding noise process for all diffusion steps can be rewritten as below

$$Y_t \mid Y_{t-1}, P_{\theta}(X) = \sqrt{1-\beta_t} Y_{t-1} + (1 - \sqrt{1-\beta_t}) P_{\theta}(X) + \epsilon,$$
$$\epsilon \sim \mathcal{N}(\mathbf{0}, \beta_t \mathbf{I}). \tag{12}$$

Because of the additivity of the Gaussian distribution, we can present the closed form of the relationship between the original data and the data at each diffusion step like this

$$Y_t \mid Y_0, P_{\theta}(X) = \sqrt{\bar{\alpha}_t} Y_0 + (1 - \sqrt{\bar{\alpha}_t}) P_{\theta}(X) + \bar{\epsilon},$$
$$\bar{\epsilon} \sim \mathcal{N}(\mathbf{0}, (1 - \bar{\alpha}_t)\mathbf{I}). \tag{13}$$

Note that this formula shows that the forward process is a process of gradually interpolating from the real value to the predicted one, and ultimately becoming the predicted one.

Similar to DDPM [35] and [33], we can break the optimized target into several parts. To get the $\mathcal{L}_{t-1}(Y_0, X)$, we need to calculate the reverse process $q(Y_{t-1} \mid Y_t, Y_0, P_{\theta}(X))$, which can be represented by a normal distribution $\mathcal{N}(Y_{t-1}; \tilde{\boldsymbol{\mu}}(Y_t, Y_0, P_{\theta}(X)), \tilde{\beta}_t \mathbf{I})$ as below

$$q(Y_{t-1} \mid Y_t, Y_0, P_{\theta}(X)) \tag{14}$$
$$= q(Y_t \mid Y_{t-1}, Y_0, P_{\theta}(X)) \frac{q(Y_{t-1} \mid Y_0, P_{\theta}(X))}{q(Y_t \mid Y_0, P_{\theta}(X))}, \tag{15}$$
$$= q(Y_t \mid Y_{t-1}, P_{\theta}(X)) \frac{q(Y_{t-1} \mid Y_0, P_{\theta}(X))}{q(Y_t \mid Y_0, P_{\theta}(X))}, \tag{16}$$
$$\propto \exp\left(-\frac{1}{2}\left(\frac{(Y_t - \sqrt{\alpha_t} Y_{t-1} - (1 - \sqrt{\alpha_t}) P_{\theta}(X))^2}{\beta_t}\right.\right.$$
$$+ \frac{(Y_{t-1} - \sqrt{\bar{\alpha}_{t-1}} Y_0 - (1 - \sqrt{\bar{\alpha}_{t-1}}) P_{\theta}(X))^2}{1 - \bar{\alpha}_{t-1}}$$
$$\left.\left. - \frac{(Y_t - \sqrt{\bar{\alpha}_t} Y_0 - (1 - \sqrt{\bar{\alpha}_t}) P_{\theta}(X))^2}{1 - \bar{\alpha}_t}\right)\right). \tag{17}$$

The third term on the right side is independent of the Y_{t-1} and we can rewrite the formula as

$$q(Y_{t-1} \mid Y_t, Y_0, P_{\theta}(X)) \tag{18}$$
$$\propto \exp\left(-\frac{1}{2}\left(\frac{\alpha_t Y_{t-1}^2 - 2\sqrt{\alpha_t}(Y_t - (1 - \sqrt{\alpha_t}) P_{\theta}(X)) Y_{t-1}}{\beta_t}\right.\right.$$
$$\left.\left. + \frac{Y_{t-1}^2 - 2(\sqrt{\bar{\alpha}_{t-1}} Y_0 + (1 - \sqrt{\bar{\alpha}_{t-1}}) P_{\theta}(X)) Y_{t-1}}{1 - \bar{\alpha}_{t-1}}\right)\right). \tag{19}$$

Finally, we can write this formula in the form of a normal distribution density by completing the square like this

$$q(Y_{t-1} \mid Y_t, Y_0, P_{\theta}(X)) \propto \exp\left(-\frac{(Y_{t-1} - \tilde{\boldsymbol{\mu}}_t(Y_t, Y_0, P_{\theta}(X)))^2}{2\tilde{\beta}_t}\right), \tag{20}$$

where

$$\tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t, \tag{21}$$

TABLE 2 Hyperparameters of the FEDformer.

Modes	mode_select	Version	moving_avg	L	Base	cross_activation	Dropout	Activation
32	Random	Fourier(Wavelets)	[12, 24]	1	Legendre	tanh	0.05	gelu

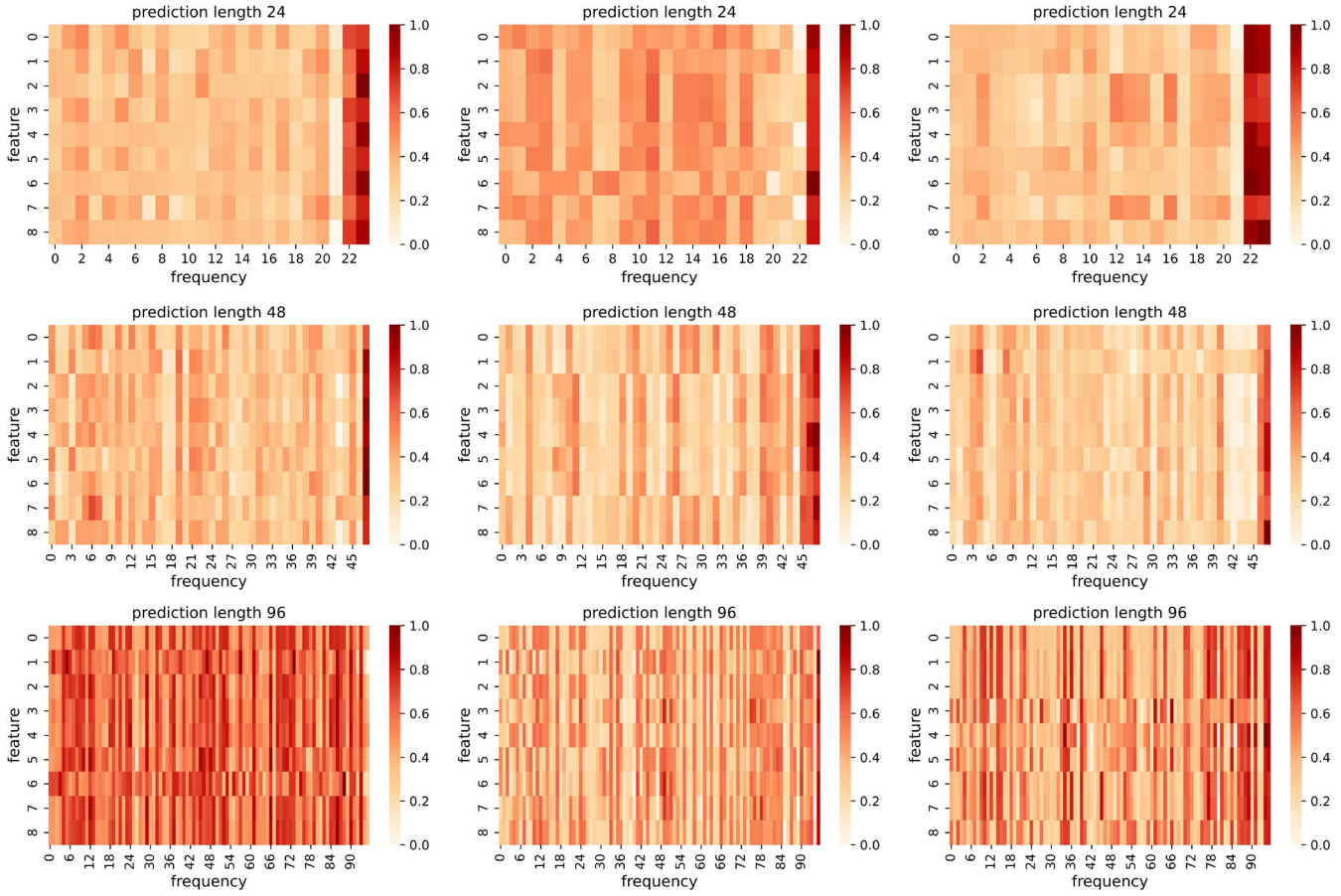

FIGURE 5 Visualization of frequency attention under different forecasting time scales.

$$\begin{aligned}
&\tilde{\boldsymbol{\mu}}_t(Y_t, Y_0, P_\theta(X)) \\
&= \underbrace{\frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}\beta_t}_{\gamma_0} Y_0 + \underbrace{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\sqrt{\alpha_t}}_{\gamma_1} Y_t \\
&+ \underbrace{\left(1 + \frac{(\sqrt{\bar{\alpha}_t}-1)(\sqrt{\alpha_t}+\sqrt{\bar{\alpha}_{t-1}})}{1-\bar{\alpha}_t}\right)}_{\gamma_2} P_\theta(X). \qquad (22)
\end{aligned}$$

In this way, the optimized target will be transformed into minimizing the distance between distribution expectations $\|\tilde{\boldsymbol{\mu}}_t(Y_t, Y_0, P_\theta(X)) - \boldsymbol{\mu}_\phi(Y_t, Y_0, P_\theta(X), t)\|^2$. Also, similar to DDPM, we can replace the expectation by the noise and our final loss function will be

$$\left\|\epsilon - \epsilon_\phi(X, \sqrt{\bar{\alpha}_t}Y_0 + \sqrt{1-\bar{\alpha}_t}\epsilon + (1-\sqrt{\bar{\alpha}_t})P_\theta(X), P_\theta(X), t)\right\|^2. \qquad (23)$$

With the label used above, we summarize our diffusion transformation framework's training and inference process in Algorithms 1 and 2, respectively. And the neural network in Figure 4 is used to create the corresponding noise during training.

4 | EXPERIMENT

4.1 | Dataset

We conduct our experiments on six wind power plants in one province of China. Because of data privacy, we normalize the data into [0,1]. Each of them contains the wind power

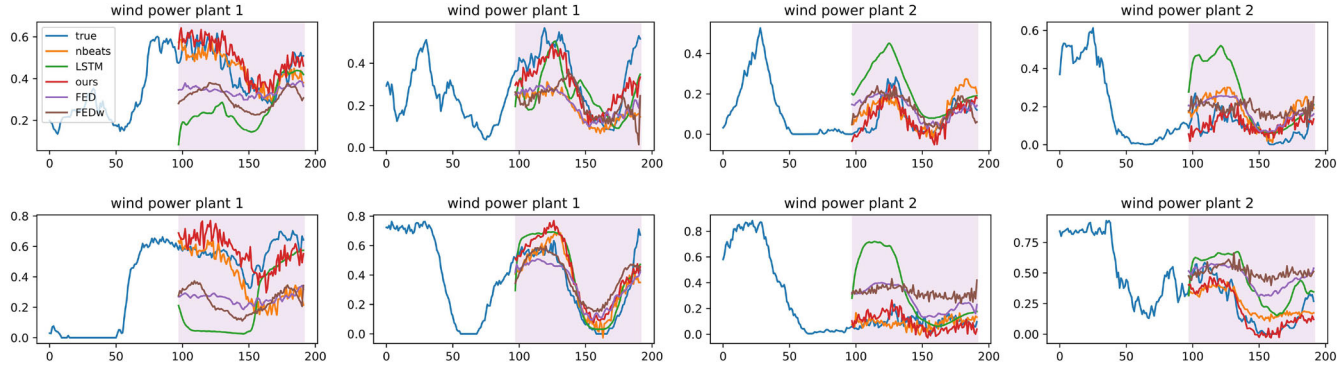

FIGURE 6 Deterministic results on wind power plant 1,2.

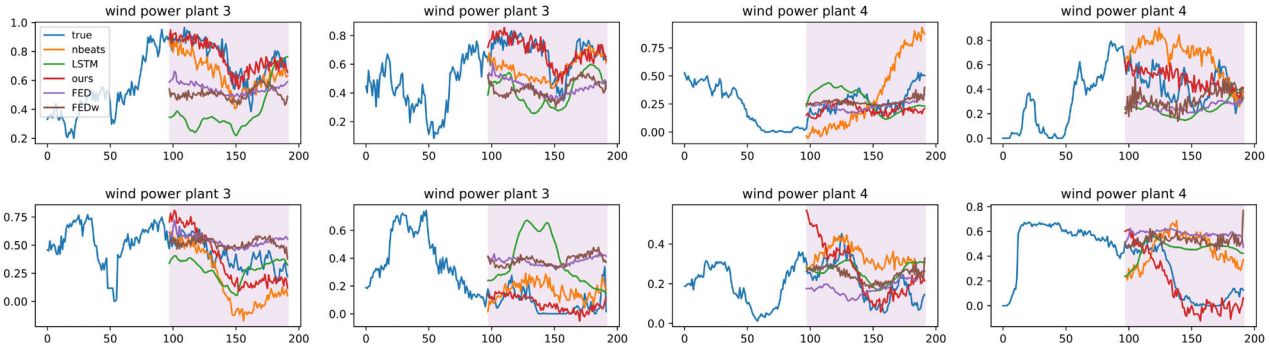

FIGURE 7 Deterministic results on wind power plant 3,4.

generation data from 2019/01/01 to 2021/12/31 with a resolution of 15 min. The training set ranges from 2019/01/01 to 2021/09/30 and we take the last three months as our test set.

4.2 | Experiment setup

Our method is based on RNN neural networks and can simultaneously give advanced deterministic and probabilistic results. Therefore, we choose several widely used deterministic and probabilistic forecasting methods based on RNN structures as our benchmarks and some widely used metrics on both deterministic forecasting and probabilistic forecasting to evaluate our methods.

4.2.1 | Benchmarks

We introduce four advanced methods to compare with our methods.

- LSTM [30] a kind of RNN structure, which is widely used in sentence modeling.
- N-BEATS [37] a deep neural architecture based on a very deep stack of fully-connected layers. Each layer can use either a general fully connected layer directly or output the coefficients of certain specific functions (such as trend functions, seasonal functions etc.), which can break the sequence into different explainable terms.
- FEDformer [29] FEDformer is a time series forecasting model based on the Transformer structure, which uses attention mechanisms in the frequency domain to reduce the impact of noise on the model in the temporal domain. It uses two frequency domain analysis methods and they are Fourier transform and wavelet transform.
- DeepAR [38] a kind of deep state model based on the RNN module, which uses the normal distribution to model the output of the deep neural network. Apart from the Gaussian distribution, we also use the student-t distribution as the emission head.
- Dropout [39] Dropout has been proven to be capable of acting as a Bayesian approximation to represent the uncertainty in deep learning. In this paper, we use the LSTM network as the basic network and use the Dropout function provided by Pytorch to capture the uncertainty and give probabilistic results.

Among them, we will compare the deterministic reduces with LSTM and N-BEATS while comparing the probabilistic results with the DeepAR and Dropout methods.

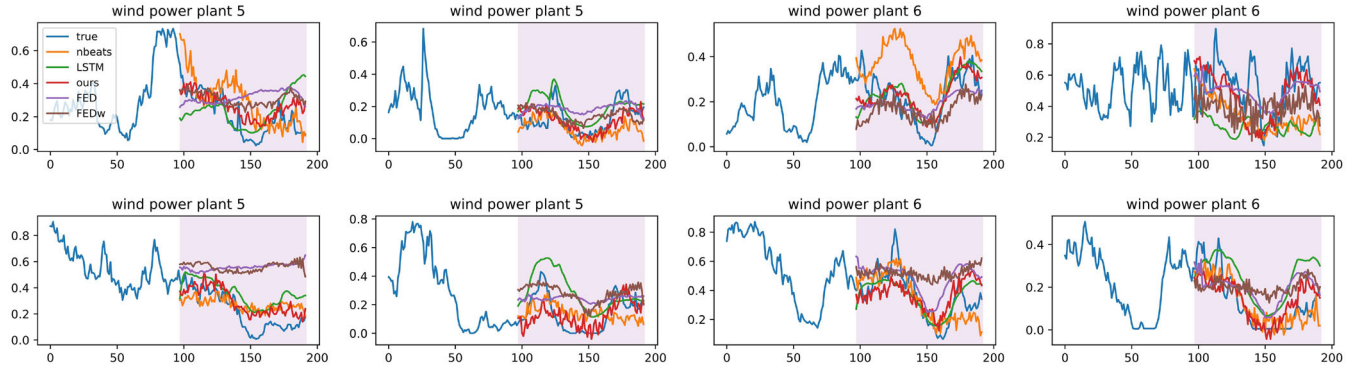

FIGURE 8 Deterministic results on wind power plant 5,6.

4.2.2 | Metric

To evaluate the deterministic result, we use Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). The MAE is the mean of the absolute errors between the predicted values and the true values. And the RMSE is the square root of the square error between the predicted value and the true value. Both of them are widely used in the regression problem and their definitions are shown below

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_t - \hat{y}_t|, \tag{24}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_t - \hat{y}_t)^2}, \tag{25}$$

where $y_t, \hat{y}_t$, and n indicate the real value, predicted value, and the number of predicted points individually.

To evaluate the probabilistic results, we also introduce two evaluation metrics. The first one is the Continuous Ranked Probability Score (CRPS) [40]. With the predicted cumulative distribution function (CDF) F_t and the real value y_t, the CRPS can be defined as follow.

$$\mathrm{CRPS}(F_t, y_t) = \int_{\mathbb{R}} (F_t(z) - \mathbb{I}\{y_t \le z\})^2 \, \mathrm{d}z, \tag{26}$$

where $\mathbb{I}\{y_t \le z\}$is the indicator function which is one if $y_t \le z$ and zero otherwise. Since not all the methods can produce the CDF, we use the samples generated from the different methods and replace the CDF with empirical CDF.

The other is the Winkler Score (WS), which is a metric for evaluating the prediction intervals (PI). For a central (1- α)% PI, it is defined as follows [41]:

$$WS_{\alpha,t} = \begin{cases} \delta, & L_t \le y_t \le U_t \\ \delta + \dfrac{2(y_t - U_t)}{2\left(\frac{\alpha}{t}\right)}, & y_t > U_t \\ \delta + \dfrac{2(L_t)}{\alpha}, & y_t < L_t \end{cases}, \tag{27}$$

where L_t and U_t represent the lower and upper bound of the PI, respectively; $\delta = U_t - L_t$. In this work, we will evaluate our methods on the 50% and 80% intervals.

4.2.3 | Hyperparameters and forecasting settings

Focusing on short-term wind power forecasting, we use historical consecutive 96, 48, and 24 data points (24h, 12h, 6h) as model input and forecast the future 96, 48, and 24 data points (24h, 12h, 6h). This forecasting setting is also comparable with the benchmarks [42, 43].

Each model is trained by ADAM with an initial learning rate of 10e-3 and the batch size is 64. Apart from the N-BEATS, all the methods are based on the LSTM module. Therefore, we use the same LSTM setting for all the methods. The hyperparameters of LSTM are listed in Table 1.

All the experiments are implemented in PyTorch and conducted on a single NVIDIA GeForce RTX 3080Ti GPU.

For FEDformer, we use the same hyperparameter settings consistent with the original paper, where $d\_ff = 16$, $d\_model = 16$, and $n\_heads = 8$. The layers of the encoder and decoder are 2 and 1, respectively. In addition, due to the different resolutions of the input data, we have added the feature of *minute_of_hour* to the default time feature in FEDformer. The rest of the hyperparameter settings for FEDformer can be seen in Table 2.

For the N-BEATS, we also set two hidden layers and they are trend blocks and seasonal blocks. For our diffusion framework, we use the same setting as [33] but reduce the steps of adding noise to 100. Experiments show that 100 is enough for our framework to perform better than competitors while the computing time reduces significantly.

4.3 | Frequency attention under different forecasting time scales

Figure 5 visualizes the frequency attention weight under different forecasting time scales. Among them, each column represents a wind power dataset. From top to bottom, they

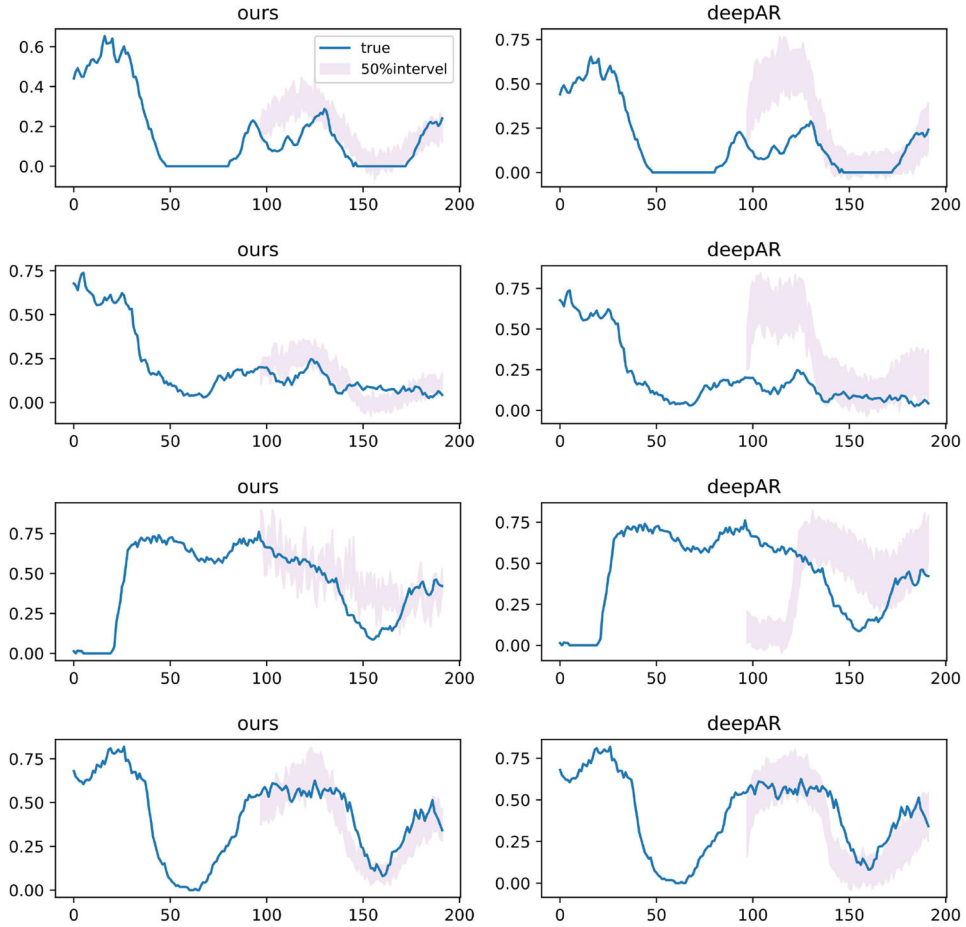

FIGURE 9 The probability forecasting results on 50% interval.

represent the predicted time scales of 6 h, 12 h, and 24 h, respectively.

From it, we can see that when the predicted time scale is 6 h (24 data points), the network's attention is mainly focused on the high-frequency information of the feature after DCT transformation. From the perspective of forecasting time length, it is understandable that when the forecasting interval is short, the network is more inclined to want to know the relationship between data points with shorter intervals. As the forecasting time interval increases, the weight of the attention layer gradually evolves from focusing on high-frequency information to an average distribution, indicating that the network not only wants to know the relationship between two data points with short intervals but also wants to know the relationship between data points with longer intervals.

4.4 | Deterministic results

Tables 3 and 4 summarize the deterministic forecasting results of our model and other models, where *I_* represents the improvement between our model and other models. For RMSE, our model exhibits superior performance over different datasets and time scales. From the average perspective, our model is also superior to other models, with improvements of 13.67%, 13.71%, and 7.17% compared to the suboptimal model on three-time scales, respectively. For MAE, the superiority of our model has been maintained and our model has improvements of 12.67%, 11.97%, and 4.93% on different time scales.

Figures 6, 7, and 8 show the 96-point (day ahead) forecasting results on six different wind power plants. We use transparent purple to distinguish between historical data input to the network and predicted one.

From them, we can see that a simple LSTM network cannot model wind power well and exhibits low performance in multiple scenarios. More specifically, the LSTM network is unable to capture the correct trend of wind power sequences. This is because the LSTM network can only capture the time-domain characteristics of wind power sequences. From Figures 6 and 7, it is evident that the forecasting results provided by LSTM are highly similar to the input sequence. This result indicates that LSTM is not able to effectively learn the features of wind power sequences and tends to provide similar results. What's more, this also indicates that relying solely on time-domain information cannot effectively model and predict wind power sequences. Apart from it, the powerful sequence modeling

TABLE 3 RMSE comparison results.

RMSE						
Data-set	Model	Ours	LSTM	N-BEATS	FED former	FED former_w
	24	**16.50**	24.29	18.03	18.77	18.01
	48	**20.77**	26.94	26.82	26.25	23.57
1	96	**24.99**	29.51	26.86	28.23	28.95
	24	**22.51**	29.55	26.14	25.63	24.73
	48	**29.51**	36.90	31.39	36.82	36.91
2	96	**36.79**	41.93	39.98	40.16	42.21
	24	**4.47**	5.71	5.36	4.74	5.02
	48	**5.69**	7.13	6.90	6.39	5.99
3	96	**7.33**	7.68	7.69	8.09	8.17
	24	**5.43**	7.20	6.45	6.21	5.53
	48	**6.94**	8.14	8.26	7.93	7.53
4	96	8.89	9.13	9.56	**8.87**	9.15
	24	**5.44**	6.60	6.31	5.78	6.48
	48	**6.67**	7.80	7.99	7.37	7.15
5	96	**8.18**	8.86	8.51	8.99	8.86
	24	**5.79**	7.59	7.37	6.74	6.54
	48	**7.86**	8.47	8.38	8.40	8.74
6	96	**8.96**	9.54	9.90	9.64	9.91
	24	**10.02**	13.49	11.61	11.31	11.05
	48	**12.90**	15.90	14.96	15.53	14.98
mean	96	**15.86**	17.77	17.08	17.33	17.88

TABLE 4 MAE comparison results.

MAE						
Data-set	Model	Ours	LSTM	N-BEATS	FED former	FED former_w
	24	**11.86**	18.11	12.80	13.59	12.86
	48	**14.90**	20.65	18.54	20.02	17.66
1	96	**18.60**	22.60	19.17	21.24	21.89
	24	**15.27**	21.66	17.91	17.52	16.81
	48	**21.28**	27.53	22.42	26.70	26.89
2	96	**26.23**	31.99	28.34	28.64	30.56
	24	3.37	4.39	3.79	**3.34**	3.64
	48	**4.21**	5.65	5.18	4.83	4.37
3	96	**5.56**	6.37	5.89	6.36	6.45
	24	**3.83**	5.78	4.53	4.63	3.88
	48	**5.27**	6.66	5.97	5.85	5.64
4	96	7.16	7.50	7.26	**6.70**	6.90
	24	**3.90**	4.94	4.34	4.18	4.88
	48	**4.81**	5.91	5.95	5.15	5.00
5	96	**5.91**	6.84	6.02	6.44	6.35
	24	**4.18**	5.93	5.18	4.88	4.80
	48	**5.91**	6.70	6.00	6.27	6.62
6	96	**6.67**	7.44	7.10	7.15	7.47
	24	**6.06**	8.69	6.94	6.87	6.69
	48	**8.05**	10.44	9.15	9.83	9.45
mean	96	**10.02**	11.82	10.54	10.93	11.37

model N-BEATS can relatively capture temporal information well, achieving better forecasting results than LSTM. Compared with LSTM which solely considers the time-domain features of wind power sequences, N-BEATS uses carefully designed trend and seasonal blocks and adopts a residual structure for multi-step fitting. The figures show that N-BEATS can basically fit the trend of wind power series changes correctly, rather than simply outputting predicted values similar to the input. However, this approach also has limitations. The well-designed block by N-BEATS enhances the interpretability of the model, but at the same time reduces its modeling ability of the model. From Figures 6, 7, and 8, it can be seen that the initial forecasting results have a significant impact on the overall forecasting. When the initial forecasting results are good (in the second row and second column of Figure 6, and in the second row and fourth column of Figure 8), N-BEATS usually gives good forecasting results. However, when the initial offset is large (in the first row and third column, second row, and fourth column in Figure 7), N-BEATS will also perform poorly in subsequent forecasting results due to the influence of modeling ability. Apart from it, N-BEATS lacking frequency domain information sometimes cannot keep up with sudden changes in wind power(as depicted in Figure 7's second row and third column).

As for the frequency domain forecasting model FEDformer information, we compared the Fourier transform-based and wavelet transform-based FEDformer, denoted as *FED* and *FEDw*, respectively. Among them, whether using Fourier transform or wavelet transform, the forecasting results of FEDformer show a straight line shape in multiple places, indicating that FEDformer cannot fit wind power data well. On the one hand, Transformer's modeling ability for time series is often criticized [44], while on the other hand, FEDformer decomposes time series in hidden space first. Unlike N-BEATS' residual-based decomposition, the decomposition in the hidden space relies on the autocorrelation of the sequence itself, while the autocorrelation of wind power is not significant. These may be the reasons why FEDformer performs poorly in wind power sequences.

When it comes to our model, we take into account the frequency domain information of the wind power, thereby improving the expression ability of the model. On the one hand, in the wind power sequence, due to the influence of meteorological factors, the autocorrelation of the different segments of the same sequence is often small, that is, the similarity between different segments in the time domain is relatively small. Our model extracts and learns the frequency domain information of the sequence, so it will not provide similar forecasting results in the time domain like LSTM. On the other hand, since frequency information is considered, our model is less affected by the initial predicted output values. From the second row and fourth

TABLE 5 CRPS comparison results.

CRPS							
Dataset	Model	Ours	deepAR	deepAR (student-T)	Dropout (0.25)	Dropout (0.1)	I_deepAR
1	24	**8.62**	12.45	12.12	16.32	16.60	30.72%
	48	**11.26**	15.12	16.00	19.48	19.11	25.48%
	96	**14.18**	16.69	17.39	21.66	21.82	15.03%
2	24	**11.55**	16.13	15.91	19.35	20.37	28.40%
	48	**15.51**	20.02	20.33	25.53	25.52	22.51%
	96	**20.32**	23.75	24.78	31.39	31.03	14.43%
3	24	**2.43**	3.22	3.30	4.05	4.09	24.55%
	48	**3.15**	4.06	4.41	5.18	5.31	22.30%
	96	**4.13**	4.51	4.74	6.03	6.12	8.46%
4	24	**2.86**	4.04	3.92	5.12	5.15	29.25%
	48	**3.78**	4.86	4.47	6.25	6.19	22.29%
	96	5.02	5.10	**4.92**	7.11	7.45	1.46%
5	24	**2.84**	3.51	3.51	4.45	4.40	19.15%
	48	**3.55**	4.17	4.05	5.25	5.25	14.98%
	96	**4.42**	4.56	4.62	6.28	6.05	3.06%
6	24	**3.09**	4.25	4.34	5.37	5.44	27.38%
	48	**4.27**	4.79	4.94	6.14	6.31	10.79%
	96	**5.03**	5.13	5.32	6.89	7.04	1.91%
mean	24	**5.23**	7.27	7.18	9.11	9.34	28.02%
	48	**6.92**	8.83	9.03	11.30	11.28	21.67%
	96	**8.85**	9.95	10.29	13.22	13.25	11.10%

TABLE 6 Winker Score(50%) comparison results.

Winker Score (50%)							
Dataset	Model	Ours	deepAR	deepAR (student-T)	Dropout (0.25)	Dropout (0.1)	I_deepAR
1	24	**9.66**	13.84	13.47	16.97	17.20	30.23%
	48	**12.48**	16.76	17.81	20.19	19.71	25.56%
	96	**15.57**	18.48	19.27	22.06	22.09	15.74%
2	24	**12.91**	17.88	17.67	20.06	21.00	27.78%
	48	**17.19**	22.21	22.54	26.25	26.22	22.59%
	96	**22.38**	26.18	27.39	32.17	31.50	14.50%
3	24	**2.73**	3.58	3.65	4.18	4.21	23.79%
	48	**3.52**	4.55	4.89	5.34	5.46	22.52%
	96	**4.59**	5.06	5.28	6.14	6.25	9.33%
4	24	**3.22**	4.48	4.34	5.33	5.34	28.09%
	48	**4.24**	5.40	4.95	6.46	6.38	21.42%
	96	5.62	5.66	**5.47**	7.25	7.56	0.74%
5	24	**3.17**	3.89	3.88	4.62	4.56	18.31%
	48	**3.97**	4.62	4.48	5.43	5.41	14.13%
	96	**4.90**	5.06	5.13	6.41	6.13	3.23%
6	24	**3.46**	4.71	4.84	5.57	5.63	26.63%
	48	**4.78**	5.32	5.51	6.32	6.48	10.14%
	96	**5.61**	5.72	5.94	7.00	7.11	2.06%
mean	24	**5.86**	8.06	7.98	9.46	9.66	27.34%
	48	**7.70**	9.81	10.03	11.67	11.61	21.53%
	96	**9.78**	11.03	11.41	13.51	13.44	11.34%

column of Figure 7, it can be seen that despite the significant error between the initial predicted values and the true values, our model can also achieve better results in subsequent forecastings. This indicates that compared to the fixed block N-BEATS, frequency information reduces the negative impact of incorrect forecastings on subsequent forecastings, thereby improving the model's modeling ability.

4.5 | Probabilistic results

We evaluate our probabilistic forecasting results by CRPS, 50% Winkler Score, and 80% Winkler Score and they represent the overall situation, general situation, and extreme situation of probability forecasting. Tables 5, 6, and 7 demonstrate that our strategy consistently outperforms other approaches. In comparison to the suboptimal strategy, our approach is 28.02%, 21.67%, and 11.1% better for CRPS. The benefits of our methodology are maintained for a 50% Winkler Score, and it is 27.34%, 21.53%, and 11.34% superior. This is consistent with the findings in Figure 9, which demonstrate our technique's overall (50%) superiority to the baseline method.

When it comes to the 80% Winkler score considering extreme situations, our model still exhibits advantages over the deepAR model based on Gaussian distribution in 96-point (day-ahead) forecasting, but the advantage has decreased. This may be because our method gives a thinner interval compared to the Gaussian distribution (similar to Figure 9), which results in a greater penalty. Nevertheless, our method still outperforms other methods in terms of average.

We also compare both the deterministic forecasting results and probabilistic results with the diffusion transformation model based on standard Gaussian distribution. As mentioned above, our model may prefer a thinner interval compared to other models. To explain this phenomenon, Table 8 shows the comparison between different prior knowledge. Here $P_\theta(X)$ is our point forecasting result while $F(X)$ and $Fw(X)$ represent the FEDformer with Fourier and wavelets transformation, respectively. In most of situations, using a prior distribution based on point forecasting leads to a better result. Only in the sixth dataset, our model performs worse than the standard Gaussian distribution, this is understandable since the point forecasting result on this dataset is relatively poor when compares with other datasets (as shown in Figure 8). The probabilistic forecasting based on FEDformer as a prior is consistent with the point forecasting result, which falls behind our model comprehensively. Compared with the prior of 0, the forecasting results of FEDformer are closer to the true results. At the same time, our method is also closer to the true value than FEDformer. However, in the sixth dataset, the 0 prior

TABLE 7 Winker Score (80%) comparison results.

Winker score (80%)							
Data-set	Model	Ours	deep AR	deepAR (student-T)	Dropout (0.25)	Dropout (0.1)	I_deep AR
1	24	**14.90**	21.27	21.20	37.34	38.45	29.96%
	48	**20.24**	25.09	27.10	45.07	44.80	19.31%
	96	**26.23**	28.84	31.13	52.13	53.21	9.04%
2	24	**20.29**	26.91	27.02	44.53	47.72	24.61%
	48	**27.19**	33.03	35.26	60.06	60.18	17.68%
	96	**37.45**	40.53	46.36	74.84	75.33	7.61%
3	24	**4.13**	5.21	5.48	9.43	9.66	20.78%
	48	**5.43**	6.40	7.80	12.07	12.55	15.13%
	96	7.22	**6.75**	7.40	14.50	14.72	−6.95%
4	24	**4.75**	6.33	6.54	11.69	11.88	25.01%
	48	**6.06**	7.46	7.21	14.60	14.54	18.70%
	96	8.34	7.72	**7.64**	17.00	18.08	−8.04%
5	24	**4.70**	5.54	5.68	10.21	10.15	15.17%
	48	**5.81**	6.70	7.03	12.10	12.36	13.31%
	96	**7.80**	7.52	8.10	14.99	14.68	−3.71%
6	24	**5.21**	6.91	7.00	12.38	12.65	24.54%
	48	**7.02**	7.67	8.13	14.43	14.96	8.38%
	96	8.83	**8.31**	9.22	16.64	17.20	−6.23%
Mean	24	**9.00**	12.03	12.15	20.93	21.75	25.21%
	48	**11.96**	14.39	15.42	26.39	26.56	16.89%
	96	**15.98**	16.61	18.31	31.68	32.20	3.82%

TABLE 8 Performance comparison with different prior distribution.

Dataset	prior	RMSE	MAE	CRPS	Winker Score (50%)	Winker Score (80%)
	$\mathcal{N}(P_\theta(X), \mathbf{I})$	**20.86**	**15.00**	**11.35**	**12.57**	**20.46**
	$\mathcal{N}(0, \mathbf{I})$	21.73	16.17	12.26	13.61	21.66
	$\mathcal{N}(F(X), \mathbf{I})$	26.27	20.21	15.03	16.86	25.37
1	$\mathcal{N}(Fw(X), \mathbf{I})$	23.43	17.57	13.10	14.64	22.38
	$\mathcal{N}(P_\theta(X), \mathbf{I})$	**29.53**	**20.88**	**15.79**	**17.50**	**28.31**
	$\mathcal{N}(0, \mathbf{I})$	30.43	22.31	16.72	18.58	28.84
	$\mathcal{N}(F(X), \mathbf{I})$	35.24	25.96	19.39	21.69	33.06
2	$\mathcal{N}(Fw(X), \mathbf{I})$	34.21	25.09	18.75	20.98	31.94
	$\mathcal{N}(P_\theta(X), \mathbf{I})$	**5.82**	**4.33**	**3.24**	**3.61**	5.59
	$\mathcal{N}(0, \mathbf{I})$	5.85	4.50	3.32	3.70	**5.55**
	$\mathcal{N}(F(X), \mathbf{I})$	6.60	5.16	3.77	4.22	6.01
3	$\mathcal{N}(Fw(X), \mathbf{I})$	6.41	4.97	3.63	4.06	5.76
	$\mathcal{N}(P_\theta(X), \mathbf{I})$	**7.09**	**5.39**	**3.88**	**4.36**	6.38
	$\mathcal{N}(0, \mathbf{I})$	7.11	5.43	3.89	4.37	**6.32**
	$\mathcal{N}(F(X), \mathbf{I})$	7.60	5.70	4.18	4.68	6.78
4	$\mathcal{N}(Fw(X), \mathbf{I})$	7.17	5.29	3.89	4.36	6.41
	$\mathcal{N}(P_\theta(X), \mathbf{I})$	**6.74**	**4.92**	**3.60**	**4.01**	**6.10**
	$\mathcal{N}(0, \mathbf{I})$	7.04	5.14	3.79	4.23	6.43
	$\mathcal{N}(F(X), \mathbf{I})$	7.30	5.33	3.89	4.34	6.39
5	$\mathcal{N}(Fw(X), \mathbf{I})$	7.26	5.31	3.83	4.28	6.34
	$\mathcal{N}(P_\theta(X), \mathbf{I})$	7.62	5.64	4.13	4.62	7.02
	$\mathcal{N}(0, \mathbf{I})$	**7.45**	**5.63**	**4.09**	**4.57**	**6.84**
	$\mathcal{N}(F(X), \mathbf{I})$	8.31	6.25	4.57	5.13	7.44
6	$\mathcal{N}(Fw(X), \mathbf{I})$	8.31	6.26	4.59	5.16	7.37
	$\mathcal{N}(P_\theta(X), \mathbf{I})$	**12.94**	**9.36**	**7.00**	**7.78**	**12.31**
	$\mathcal{N}(0, \mathbf{I})$	13.27	9.86	7.35	8.18	12.61
	$\mathcal{N}(F(X), \mathbf{I})$	15.22	11.43	8.47	9.49	14.18
Mean	$\mathcal{N}(Fw(X), \mathbf{I})$	14.46	10.75	7.97	8.91	13.37

achieved the best performance, indicating that there may be a threshold where point forecasting results can provide positive assistance to probabilistic forecasting results when the threshold is exceeded. Apart from it, the metric where our model fails to achieve optimal performance is concentrated on the high quantile. This shows that our model's preference for a thinner interval may be because of prior knowledge. Even though taking relatively bad point-point forecasting may harm our model, the diffusion-based transformation doesn't rely on a specific forecasting framework, we can treat this diffusion framework as a plug-in and add it to any other advanced forecasting models to produce advanced probabilistic results.

5 | CONCLUSION

In this paper, we discuss both deterministic and probabilistic wind power forecasting. For deterministic forecasting, we introduce discrete cosine transform to capture the frequency domain features of wind power. We demonstrated that as the predicted time scale increases, the attention of neural networks will gradually shift from high-frequency information to the overall picture. For probabilistic forecasting, we use conditional diffusion models based on advanced deterministic forecasting to obtain advanced probabilistic forecasting. Compared to the deepAR model based on Gaussian distribution, our model is more confident (manifested as a narrower prediction interval), which may be because our probabilistic prediction is based on deterministic prediction. According to our experimental results, more accurate point forecasting results usually yield better probabilistic forecasting results. However, there are exceptions to this rule. In the sixth wind power plant station, probabilistic forecasting without prior information performed better, indicating that there may be a threshold for the accuracy of point forecasting. When the accuracy is lower than that, prior information may cause the results of probabilistic forecasting to deteriorate. How to find this threshold to guide the construction of forecasting models will be our future work. What is more, even though sometimes it may lead to bad results because of the relatively poor performance of the point forecasting, this diffusion framework can act as a plug-in that can assist other advanced point forecasting methods to get advanced probabilistic results.

AUTHOR CONTRIBUTIONS

Hongqiao Peng: Conceptualization, methodology, writing - original draft. Hui Sun: Software, validation. Shuxin Luo: Validation. Zhengmin Zuo: Software, visualization. Shixu Zhang: Visualization. Zhixian Wang: Formal analysis, writing - review and editing. Yi Wang: Writing - review and editing.

ACKNOWLEDGEMENTS

This work was supported by the Power Planning Special Topic of Guangdong Power Grid Co., Ltd., titled "Research on Multi-time-space Scale Renewable Energy Output Characteristics Simulation and Typical Mode Identification Methods in Guangdong Province," under Grant 031000QQ00220002.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from Planning Research Center of Guangdong Power Grid Corporation CSG. Restrictions apply to the availability of these data, which were used under license for this study. Data are available from the authors with the permission of Planning Research Center of Guangdong Power Grid Corporation CSG.

ORCID

Zhixian Wang https://orcid.org/0009-0000-2734-5074
Yi Wang https://orcid.org/0000-0003-1143-0666

REFERENCES

1. Amezquita, H., Carvalho, P.M., Morais, H.: Wind forecast at medium voltage distribution networks. Energies 16(6), 2887 (2023)
2. Manish, S., Pillai, I.R., Banerjee, R.: Sustainability analysis of renewables for climate change mitigation. Energy Sustain. Develop. 10(4), 25–36 (2006)
3. Liu, J.: China's renewable energy law and policy: A critical review. Renew. Sustain. Energy Rev. 99, 212–219 (2019)
4. Szarka, J.: Wind power, policy learning and paradigm change. Energy Policy 34(17), 3041–3048 (2006)
5. Machado, E.P., Morais, H., Pinto, T.: Wind speed forecasting using feed-forward artificial neural network. In: Distributed Computing and Artificial Intelligence, Volume 1: 18th International Conference 18, pp. 159–168. Springer, Cham (2022)
6. Taylor, J.W., McSharry, P.E., Buizza, R.: Wind power density forecasting using ensemble predictions and time series models. IEEE Trans. Energy Convers. 24(3), 775–782 (2009)
7. Watson, E.B., Etemadi, A.H.: Modeling electrical grid resilience under hurricane wind conditions with increased solar and wind power generation. IEEE Trans. Power Syst. 35(2), 929–937 (2019)
8. Vargas, S.A., Esteves, G.R.T., Maçaira, P.M., Bastos, B.Q., Oliveira, F.L.C., Souza, R.C.: Wind power generation: A review and a research agenda. J. Cleaner Prod. 218, 850–870 (2019)
9. Soman, S.S., Zareipour, H., Malik, O., Mandal, P.: A review of wind power and wind speed forecasting methods with different time horizons. In: North American Power Symposium 2010, pp. 1–8. IEEE, Piscataway (2010)
10. Chen, P., Pedersen, T., Bak-Jensen, B., Chen, Z.: Arima-based time series model of stochastic wind power generation. IEEE Trans. Power Syst. 25(2), 667–676 (2009)
11. Liu, H., Tian, H.-Q., Li, Y.-F.: Comparison of two new arima-ann and arima-kalman hybrid methods for wind speed prediction. Appl. Energy 98, 415–424 (2012)
12. Wu, Y.-K., Hong, J.-S.: A literature review of wind forecasting technology in the world. In: 2007 IEEE Lausanne Power Tech, pp. 504–509. IEEE, Piscataway (2007)
13. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: International Conference on Machine Learning, pp. 5301–5310. PMLR, New York (2019)
14. Liu, Y., Guan, L., Hou, C., Han, H., Liu, Z., Sun, Y., Zheng, M.: Wind power short-term prediction based on lstm and discrete wavelet transform. Appl. Sci. 9(6), 1108 (2019)
15. Sun, Y., Wang, P., Zhai, S., Hou, D., Wang, S., Zhou, Y.: Ultra short-term probability prediction of wind power based on lstm network and condition normal distribution. Wind Energy 23(1), 63–76 (2020)
16. Shahid, F., Zameer, A., Muneeb, M.: A novel genetic lstm model for wind power forecast. Energy 223, 120069 (2021)
17. Wu, Q., Zheng, H., Guo, X., Liu, G.: Promoting wind energy for sustainable development by precise wind speed prediction based on graph neural networks. Renew. Energy 199, 977–992 (2022)
18. Qu, K., Si, G., Shan, Z., Kong, X., Yang, X.: Short-term forecasting for multiple wind farms based on transformer model. Energy Rep. 8, 483–490 (2022)
19. Sun, M., Feng, C., Zhang, J.: Multi-distribution ensemble probabilistic wind power forecasting. Renew. Energy 148, 135–149 (2020)
20. Yang, X., Zhang, Y., Yang, Y., Lv, W.: Deterministic and probabilistic wind power forecasting based on bi-level convolutional neural network and particle swarm optimization. Appl. Sci. 9(9), 1794 (2019)
21. Xu, P., Zhang, M., Chen, Z., Wang, B., Cheng, C., Liu, R.: A deep learning framework for day ahead wind power short-term prediction. Appl. Sci. 13(6), 4042 (2023)
22. Kahraman, A., Yang, G., Hou, P.: Wind power forecasting using lstm incorporating fourier transformation based denoising technique. In: 20th International Workshop on Large-Scale Integration of Wind Power into Power Systems as well as on Transmission Networks for Offshore Wind Power Plants (WIW 2021), vol. 2021, pp. 94–98. IET, Stevenage (2021)
23. Ye, L., Li, Y., Pei, M., Zhao, Y., Li, Z., Lu, P.: A novel integrated method for short-term wind power forecasting based on fluctuation clustering and history matching. Appl. Energy 327, 120131 (2022)
24. Jiang, M., Zeng, P., Wang, K., Liu, H., Chen, W., Liu, H.: Fecam: Frequency enhanced channel attention mechanism for time series forecasting. arXiv preprint, arXiv:2212.01209 (2022)
25. Wang, H.-Z., Li, G.-Q., Wang, G.-B., Peng, J.-C., Jiang, H., Liu, Y.-T.: Deep learning based ensemble approach for probabilistic wind power forecasting. Appl. Energy 188, 56–70 (2017)
26. Aly, H.H.: A novel deep learning intelligent clustered hybrid models for wind speed and power forecasting. Energy 213, 118773 (2020)
27. Stefenon, S.F., Seman, L.O., Aquino, L.S., dos Santos Coelho, L.: Wavelet-seq2seq-lstm with attention for time series forecasting of level of dams in hydroelectric power plants. Energy 274, 127350 (2023)
28. Lin, Y., Chen, K., Zhang, X., Tan, B., Lu, Q.: Forecasting crude oil futures prices using bilstm-attention-cnn model with wavelet transform. Appl. Soft Comput. 130, 109723 (2022)
29. Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R.: Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In: International Conference on Machine Learning, pp. 27 268–27 286. PMLR, New York (2022)
30. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: Lstm cells and network architectures. Neural Comput. 31(7), 1235–1270 (2019)
31. Ahmed, N., Natarajan, T., Rao, K.R.: Discrete cosine transform. IEEE Trans. Comput. 100(1), 90–93 (1974)
32. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141. IEEE, Piscataway (2018)
33. Han, X., Zheng, H., Zhou, M.: Card: Classification and regression diffusion models. arXiv preprint, arXiv:2206.07275 (2022)

34. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Adv. Neural Inf. Process Syst. 33, 6840–6851 (2020)
35. Nachmani, E., Roman, R.S., Wolf, L.: Denoising diffusion gamma models. arXiv preprint, arXiv:2110.05948 (2021)
36. Pandey, K., Mukherjee, A., Rai, P., Kumar, A.: Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents. arXiv preprint, arXiv:2201.00308 (2022)
37. Oreshkin, B.N., Carpov, D., Chapados, N., Bengio, Y.: N-beats: Neural basis expansion analysis for interpretable time series forecasting. arXiv preprint, arXiv:1905.10437 (2019)
38. Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T.: Deepar: Probabilistic forecasting with autoregressive recurrent networks. Int. J. Forecast. 36(3), 1181–1191 (2020)
39. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning, pp. 1050–1059. PMLR, New York (2016)
40. Matheson, J. E., Winkler, R.L.: Scoring rules for continuous probability distributions. Manage. Sci. 22(10), 1087–1096 (1976)
41. Wang, C., Qin, D., Wen, Q., Zhou, T., Sun, L., Wang, Y.: Adaptive probabilistic load forecasting for individual buildings. iEnergy 1(3), 341–350 (2022)
42. Li, Y., Lu, X., Wang, Y., Dou, D.: Generative time series forecasting with diffusion, denoise, and disentanglement. arXiv preprint, arXiv:2301.03028 (2023)
43. Rasul, K., Seward, C., Schuster, I., Vollgraf, R.: Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In: International Conference on Machine Learning, pp. 8857–8868. PMLR, New York (2021)
44. Zeng, A., Chen, M., Zhang, L., Xu, Q.: Are transformers effective for time series forecasting? In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37(9), pp. 11 121–11 128. AAAI Press, Menlo Park, CA (2023)

How to cite this article: Peng, H., Sun, H., Luo, S., Zuo, Z., Zhang, S., Wang, Z., Wang, Y.: Diffusion-based conditional wind power forecasting via channel attention. IET Renew. Power Gener. 18, 306–320 (2024). https://doi.org/10.1049/rpg2.12825