Available online at https://link.springer.com/journal/42241
http://www.jhydrodynamics.com
Journal of Hydrodynamics, 2026
https://doi.org/10.1007/s42241-026-0041-x

Addressing scale discrepancy in wind turbine wake turbulent kinetic energy predictions via quantile transformation and bayesian deep learning

Zhi-hao Tang, Mao-kun Ye, Yi-sheng Yao, De-cheng Wan*

Computational Marine Hydrodynamics Lab (CMHL), School of Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

(Received January 19, 2026, Revised February 13, 2026, Accepted February 14, 2026, Published online June 2, 2026)

Abstract: Wind energy is a crucial component of the global transition to green energy. The development of wind resources relies on wind turbines, which are typically clustered in wind farms. The wake generated by upstream turbines significantly impacts downstream turbines, leading to reduced power generation and increased fatigue loads. turbulent kinetic energy (TKE) is a key physical quantity in wind turbine wake prediction, governing wake mixing and recovery processes, thus holding substantial research importance. However, a major challenge in data-driven wake prediction is the significant scale discrepancy in TKE distribution. TKE values are predominantly low across most of the wake field, with only a few critical regions (e.g., behind the blade tips and hub) exhibiting high values. This highly skewed distribution causes conventional data-driven models to often treat these high-value points as outliers, leading to poor learning of critical features and increased model uncertainty. This study proposes a novel data-driven framework combining quantile transformation and Bayesian deep learning to accurately predict the TKE distribution in wind turbine wakes. The nonlinear quantile transformation effectively mitigates the scale discrepancy issue, enabling the neural network to learn features from the highly imbalanced data more effectively. Meanwhile, the Bayesian deep learning approach quantifies the predictive uncertainty, enhancing the model's reliability. The proposed method is validated against high-fidelity computational fluid dynamics (CFD) data, demonstrating superior performance compared with traditional scaling methods like Min-Max scaler.

Key words: Wind turbine wake, machine learning, Bayesian neural networks (BNNs)

0. Introduction

Wind power plays a pivotal role in the global shift toward sustainable energy. Within wind farms, turbines are deployed in clusters, where wake effects from upstream turbines can significantly reduce power output and increase fatigue loads on downstream turbines. Wake characteristics, particularly turbulent kinetic energy (TKE), are critical as TKE governs the wake's mixing efficiency and recovery rate. Accurate TKE prediction is therefore essential for optimizing the layout and operational performance of wind farms.

The accurate prediction of wind turbine wake characteristics is paramount for optimizing the performance and layout of wind farms, where the interaction between upstream and downstream turbines can result in significant power losses and intensified structural fatigue. Traditional methodologies for tackling this issue have primarily centered on two distinct computational paradigms: analytical models and computational fluid dynamics (CFD) simulations. Each approach provides a unique balance between computational efficiency and predictive fidelity.

Analytical wake models, grounded in simplified physical principles such as momentum conservation and simplified turbulence assumptions, offer a computationally efficient framework for rapid evaluation. Seminal models including those proposed by Jensen[1] and subsequent modifications[2-6] are widely employed in initial wind farm layout planning and annual energy production estimation due to their high computational speed and structural simplicity. However, these models suffer from a notable limitation in terms of inherent inaccuracy under complex flow regimes. They often struggle to fully capture critical phenomena such as the detailed structure of the velocity deficit, the distribution of TKE, and the complex

Project supported by National Key Research and Development Program of China (Grant No. 2025YFF0519802).
Biography: Zhi-hao Tang (2004-), Male, Bachelor,
E-mail: gimgoontang@sjtu.edu.cn
Corresponding author: De-cheng Wan,
E-mail: dcwan@sjtu.edu.cn

interactions of multiple wakes under varying atmospheric stability conditions[7-8]. Their simplified treatment of turbulence and inability to model the complete three-dimensional nature of the flow limit their suitability for detailed aerodynamic and aeroelastic analyses.

In contrast, CFD simulations provide a high-fidelity alternative through numerical solution of the governing flow equations. Approaches based on the Reynolds Averaged Navier-Stokes (RANS) equations, often coupled with two-equation turbulence models, have been extensively adopted in engineering simulations, as they strike a reasonable balance between accuracy and computational cost[9-10]. To achieve even higher fidelity, large eddy simulation resolves large energy-containing turbulent structures while modeling smaller scales, delivering superior predictions for wake meandering and complex turbulence dynamics[11-12]. Despite their high accuracy, high-fidelity CFD simulations are computationally expensive, demanding substantial computational resources and time which make them impractical for large-scale parameter studies, real-time control, or long-term operational optimization of wind farms[13].

The emergence of data-driven methods, particularly machine learning and deep learning, offers a promising alternative for bridging the gap between the computational speed of analytical models and the predictive accuracy of CFD simulations. These methods learn complex nonlinear mappings between input parameters (e.g. turbine operational states, inflow conditions, spatial coordinates) and output flow fields from high-fidelity datasets, thereby enabling rapid predictions after training[14]. Neural networks have been successfully employed for predicting wake velocity profiles[15-16]. A particularly impactful advancement lies in the integration of Bayesian deep learning, which not only yields point estimates but also quantifies predictive uncertainty, thereby providing a measure of model confidence critical for robust decision-making[17-20]. By leveraging these techniques, data-driven surrogates can attain near-real-time performance while capturing the complex physics embedded in high-fidelity data, paving new avenues for dynamic wind farm control and optimization under uncertain conditions.

These data-driven models are capable of learning complex flow patterns from high-fidelity simulation or experimental data. However, a significant challenge remains unresolved: the scale discrepancy in TKE data. The TKE distribution within a wake is highly non-uniform, with vast regions exhibiting extremely low TKE values and extremely high values concentrated in small yet critical areas. This data imbalance causes standard neural networks to prioritize the majority of low-value data points, often treating the sparse high-value points as noise to mitigate overfitting. Consequently, the model fails to accurately capture the high-TKE features that are essential for understanding wake dynamics. Furthermore, this inherent data characteristic leads to substantial predictive uncertainty in the model.

To address these challenges, this work proposes a framework integrating a quantile transformer for preprocessing and a Bayesian neural network (BNN) for prediction. The quantile transformer nonlinearly transforms TKE data into a uniform distribution, alleviating the scale discrepancy and enabling the model to effectively learn from all data regions. Implemented via Monte Carlo dropout, the BNN not only provides point predictions but also quantifies the epistemic uncertainty associated with the predictions, thereby offering a measure of model confidence.

1. Methodology

1.1 *Formulation of the problem*

The objective of this work is to develop a data-driven surrogate model that maps spatial coordinates and time to the TKE values within wind turbine wakes. This problem is formally formulated as a regression task:

$$T_{\mathrm{TKE}} = B_{\mathrm{BNN}}(x, y, z, t) \tag{1}$$

where (x, y, z) are the spatial coordinates, t is the time, and the B_{BNN} serves as the nonlinear approximator.

1.2 *Quantile transformation*

Quantile transformation is a nonparametric data preprocessing technique designed to reshape the distribution of a given variable by matching its quantiles to those of a specified target distribution, typically either a uniform or a Gaussian distribution. The core mechanism operates through a two-step mapping process. Let F_X denote the empirical cumulative distribution function (CDF) of the original random variable X, and let G represent the CDF of the desired target distribution. The transformed variable X' is then defined as:

$$X' = G^{-1}[F_X(X)] \tag{2}$$

where G^{-1} is the inverse CDF (quantile function) of the target distribution. In essence, each data point is first assigned a probability value according to its rank within the original empirical distribution, which is subsequently substituted with the corresponding quantile derived from the target distribution. This procedure ensures a monotonic relationship between the original and transformed values while fundamentally modifying the underlying distributional shape.

The primary rationale for employing quantile transformation stems from its exceptional robustness to outliers and its capacity to enforce the distributional assumptions mandated by numerous statistical models and machine learning algorithms. Unlike parametric scaling methods (e.g. standardization or Min-Max normalization) that are sensitive to extreme values, quantile transformation depends exclusively on the ordinal structure of the data. As a consequence, outliers are mapped to the tails of the target distribution without unduly influencing the transformation of the remaining data points. Furthermore, when configured to output a Gaussian distribution, this technique effectively normalizes features, thereby fulfilling the normality prerequisites of various classical modeling approaches (e.g. linear regression, linear discriminant analysis, and Gaussian processes).

1.3 *Bayesian deep learning*

A BNN employed in this work regards the network weights as probability distributions rather than deterministic scalar values. This Bayesian framework enables the quantification of predictive uncertainty, which stems from the model's inherent lack of knowledge (i.e. epistemic uncertainty).

Although both artificial neural networks (ANNs) and BNNs consist of interconnected layers of nonlinear units, they differ fundamentally in theoretical framework, optimization objectives, and predictive characteristics. A standard ANN operates as a deterministic function approximator, where training seeks a single optimal set of fixed parameters θ^* that minimize a specified loss function on the training data $D = \{(x_i, y_i)\}_{i=1}^N$, such as the common mean squared error:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N}[y_i - f_\theta(x_i)]^2 \tag{3}$$

The learned model $f_{\theta^*}(x)$ subsequently provides a point estimate for any new input x^*, offering no inherent measure of its own confidence. In stark contrast, a BNN adopts a probabilistic Bayesian perspective, treating the network parameters θ as random variables governed by a prior distribution $p(\theta)$ and aiming to infer their full posterior distribution $p(\theta|D)$ after observing the data.

Since exact computation of this posterior is intractable for complex models, practical BNNs rely on approximations, a widely-used technique, as implemented in the provided code, is Monte Carlo (MC) dropout, which interprets dropout applied during training as a variational Bayesian method that approximates the true posterior with a simpler distribution $q(\theta)$.

The core distinction manifests during prediction: while an ANN executes a deterministic forward pass, a BNN performs Bayesian model averaging by integrating over the parameter posterior to obtain the predictive distribution:

$$p(y^*|x^*, D) = \int p(y^*|z^*, \theta)p(\theta|D)\mathrm{d}\theta \tag{4}$$

The above integral is approximated practically via T stochastic forward passes (e.g. with dropout active at inference), yielding samples $\{\hat{y}^{(t)}\}_{t=1}^{\mathrm{T}}$. The final prediction is then the empirical mean:

$$E[y^*] \approx \frac{1}{T}\sum_{t=1}^{\mathrm{T}}\hat{y}^{(t)} \tag{5}$$

and the total predictive uncertainty is quantified as the standard deviation:

$$\sigma(y^*) \approx \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{y}^{(t)})^2 - (E|y^*)^2} \tag{6}$$

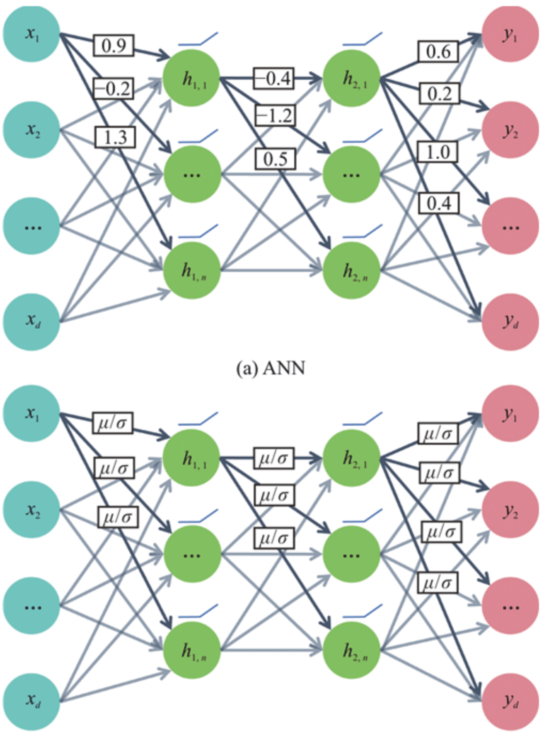
This uncertainty naturally decomposes into aleatoric uncertainty (inherent data noise) and epistemic uncertainty (model uncertainty due to limited data), the latter being particularly valuable for identifying out-of-distribution inputs or ambiguous cases. Consequently, a BNN provides not just a prediction but a calibrated measure of reliability, enabling safer deployment in risk-sensitive applications like medical diagnosis or autonomous systems, whereas a standard ANN, despite its often superior computational efficiency and simpler deployment, can produce overconfident and erroneous predictions when faced with novel or anomalous inputs. The computational overhead of BNNs, primarily from multiple inference samples, is mitigated by techniques like MC dropout, which allows approximate Bayesian inference with minimal modification to standard ANN architectures and training pipelines.

In this study, the BNN is practically realized using MC Dropout[21]. During both the training and inference phases, dropout layers remain activated. By conducting multiple forward passes (i.e. MC sampling) for a given input–with different neurons randomly deactivated in each pass–an ensemble of predictions is generated. The mean of these predictions is adopted as the final predicted TKE value, and the standard deviation serves as a quantification of the predictive uncertainty:

$$T_{\mathrm{TKE_{pred}}} = \frac{1}{N}\sum_{i=1}^{N} B_{\mathrm{BNN_{dropout}}}(x, y, z, t) \tag{7}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[B_{\mathrm{BNN_{dropout}}}(x, y, z, t) - T_{\mathrm{TKE_{pred}}}]^2} \tag{8}$$

where N is the number of MC samples, $T_{\mathrm{TKE_{pred}}}$ is the mean value of the N predictions, $B_{\mathrm{BNN_{dropout}}}$ denotes the Bayesian neural network output with dropout activated for a given input of spatial coordinates (x, y, z) and time t. The notation aligns with the input definitions in Eq. (1) and Fig. 1.



(a) ANN

(b) BNN

Fig. 1 Comparison of traditional ANNs with BNNs. (μ is the mean of the predictive distribution representing the point estimate of the TKE field, $h_{1,n}$, $h_{2,n}$ are the parameters of the BNN prior and posterior distributions, respectively)

2. Test case description

2.1 *Introduction to the wind tunnel test*

In this study, the dataset is derived from the well-known blind test workshop on wind turbine wake modeling organized by the Norwegian University of Science and Technology (NTNU). The blind test 1 (BT1) experiment featured a three-bladed wind turbine model equipped with blades designed based on the NREL S826 airfoil, which was tested in a closed-loop wind tunnel[22-24] (Fig. 2).

Fig. 2 Setup of the NTNU BT1 wind turbine[22]

The wind tunnel dimensions were 11.15 m in length, 2.71 m in width, with a height varying from 1.801 m at the inlet to 1.851 m at the outlet. The rotor diameter (D_r) was 0.897 m, positioned 3.66 m from the inlet on the tunnel's centerline and 0.817 m above the floor. The inflow velocity was constant at 10 m/s with a very low turbulence intensity of 0.3%. Different tip-speed ratios (R_{TSR}) were achieved by varying the rotational speed. Measurements included thrust coefficient (C_{T}), power coefficient (C_{P}), and wake characteristics.

2.2 *CFD simulation of wind turbine wake*

In the present study, the incompressible RANS equations are solved in the simulations:

$$\nabla \cdot \boldsymbol{V} = 0 \tag{9}$$

$$\frac{\partial \boldsymbol{V}}{\partial t} + \nabla\cdot(\boldsymbol{VV}) = \nabla\cdot\{(\nu+\nu_t)[\nabla\boldsymbol{V} + (\nabla\boldsymbol{V})^{\mathrm{T}}]\} - \nabla\left(\frac{p}{\rho} + \frac{2}{3}k\right) + \boldsymbol{B} \tag{10}$$

where $\boldsymbol{V}$ is the velocity field of mean flow, ρ is the density, p is the pressure, $\boldsymbol{B}$ is the body force vector, k is the turbulent kinetic energy, ν is the kinematic viscosity of air, and ν_t is the so-called eddy viscosity.

In machine learning tasks, a high-quality dataset serves as a fundamental prerequisite. Therefore, this chapter elaborates on the generation process of the high-quality dataset employed in this work.

2.3 *Numerical setup*

The computational domain was strategically partitioned into three regions via a multi-block approach: an inner rotating domain encompassing the rotor, a stationary wake domain extending downstream, and a stationary outer domain conforming to the wind tunnel contours. The inner rotating domain utilized unstructured hexahedral cells to accommodate the complex blade geometry, while the wake and outer domains employed structured grids. Wall boundaries (blades, hub, nacelle, tower) were resolved with viscous layers ensuring $y^+ \leq 1$. Grids were refined at critical regions like leading/trailing edges, tip, and root areas to capture relevant vortices. A systematic grid verification study was performed specifically for the wake domain.

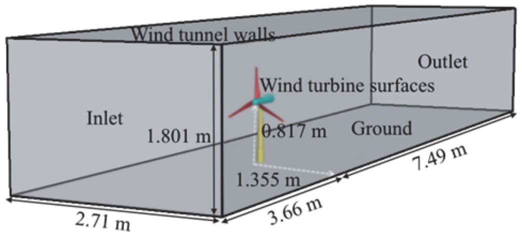


Fig. 3 Illustration of the computational domain[23]

Computational grids are constructed via the multiblock technique, with a schematic of the computational domain presented in Fig. 3. As illustrated therein, the coordinate system is defined as follows: The origin is situated at the rotor center, the positive x-axis is directed from the inlet to the outlet, the y-axis is oriented from the bottom to the top, and the z-axis is determined in accordance with the right-hand rule. The computational domain is decomposed into three distinct subdomains: (1) The inner rotating domain, a short cylinder enclosing the rotor. (2) The stationary wake domain, located downstream of the rotor and extending to the outlet. (3) The stationary outer domain, which is constructed to match the wind tunnel configuration employed in the experiment. During simulations, the inner rotating domain undergoes rotational motion, while the wake and outer domains remain stationary. These subdomains are interconnected by sliding interfaces, through which the flow field solutions of one subdomain are extrapolated to transmit flow information to adjacent subdomains. This decomposition strategy allows for the refinement of individual subdomains without modifying the others, thereby enabling a systematic verification study specifically targeting the wake region.

Two distinct grid generation techniques are implemented for the three subdomains: Unstructured hexahedral meshes are generated for the inner rotating domain, which accommodates the highly twisted blade geometry, whereas structured grids are adopted for the wake and outer domains. For the wind turbine wall boundaries–including the blades, hub, nacelle, and tower–viscous layers are constructed with the criterion of y^+ and the cell expansion ratio normal to the wall is set to 1.2. High refinement levels are applied to the leading edges, trailing edges, and other high-skew regions of the turbine blades. Additionally, the meshes in the tip and root regions are further refined to accurately capture tip and root vortices. Schematics of the computational grids for the inner rotating domain and the stationary wake domain are provided in Figs. 3, 4, respectively. It is noteworthy that in the subsequent verification study, only the grid within the wake domain is refined, while the meshes of the other subdomains remain unchanged. The grid resolution of the inner rotating domain is determined based on the authors' prior research findings.

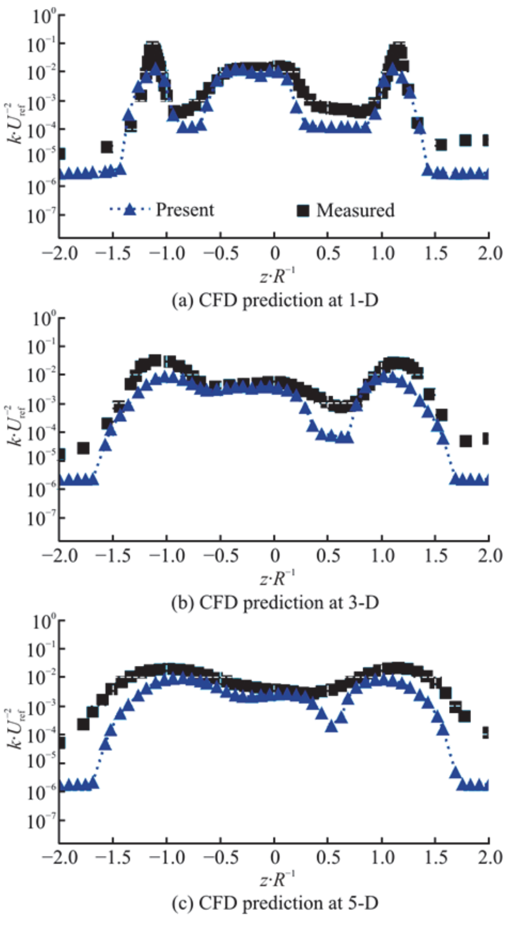


Fig. 4 CFD results of the wake field. The horizontal coordinate denotes the spanwise position normalized by rotor radius (z / R), and the vertical coordinate denotes the normalized vertical position (y / R)

2.4 *Numerical results*

The CFD methodology employed in this study has been extensively validated against experimental data in the authors' prior work[24]. The simulations effectively captured key wake characteristics, including the TKE distribution, thereby providing a high-quality dataset for subsequent machine learning tasks. Quantitative comparisons with experimental results

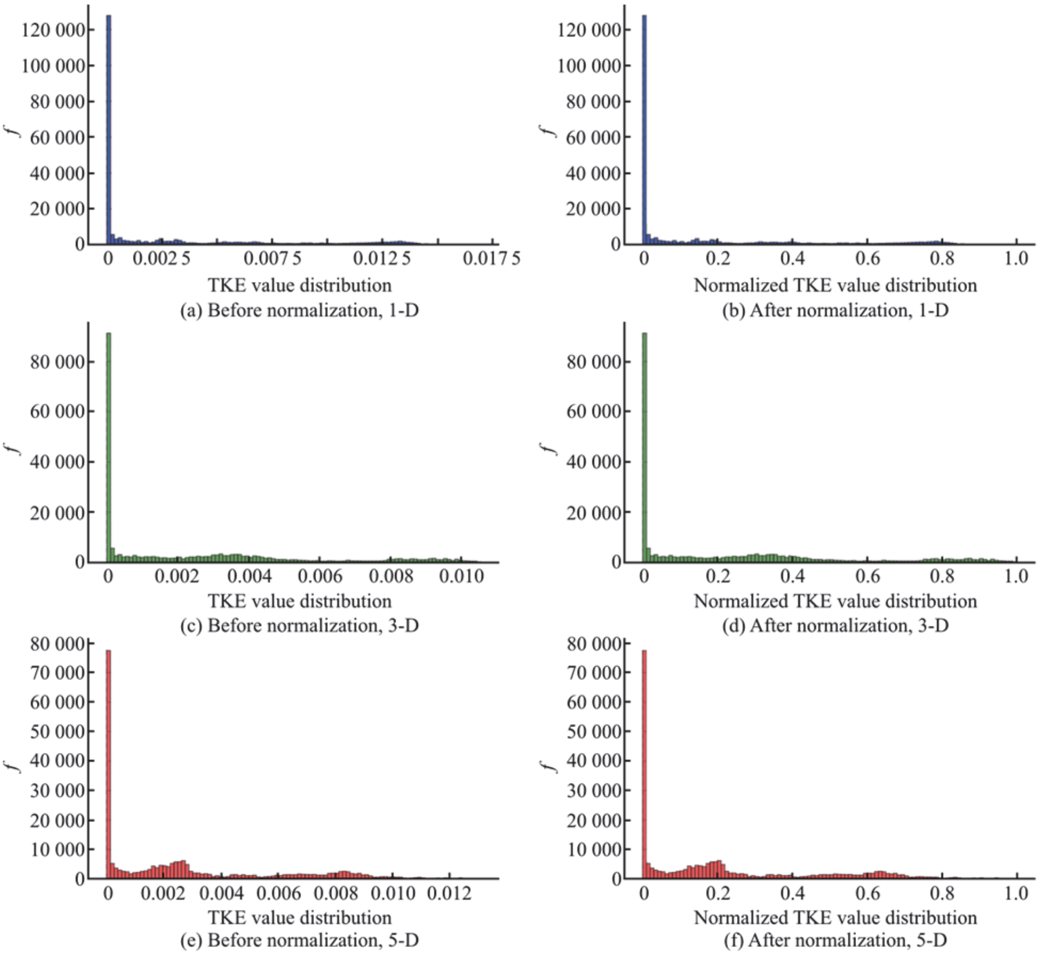


Fig. 5 Data distribution before (left-column figures) and after (right-column figures) Min-Max normalization

demonstrated good agreement, which confirms the reliability of the CFD-derived data for training and testing the neural network model.

3. Bayesian deep learning for TKE of wind turbine wake

3.1 *Preparation of dataset*

The dataset consists of flow field data from 10 rotor rotations, with each rotation sampled at 480 time steps. Velocity and TKE data were extracted along three sparse vertical lines positioned at 1-D, 3-D, and 5-D downstream of the rotor, each containing 201 data points. Data from the first 7 rotations were allocated for model training, while the remaining 3 rotations were reserved for validation and testing purposes.

To mitigate the severe scale discrepancy inherent in the TKE data, two different normalization strategies are evaluated and compared. Figure 5 illustrates the data distribution before and after the standard Min-Max normalization, where the left-column figures display the raw skewed distribution and the right-column figures show the results after Min-Max scaling. It is manifest from Fig. 5 that the traditional Min-Max normalization merely rescales the data boundaries to a $[0, 1]$ range but fails to alter the underlying probability density, the vast majority of TKE values remain excessively compressed near zero, leaving the extreme high-value critical regions poorly represented. In contrast, Fig. 6 depicts the data distribution before and after the quantile normalization. As demonstrated in the right-column figures of Fig. 6, the quantile transformation successfully maps the highly skewed empirical TKE distributions into a standard normal (or uniform) distribution. By enhancing the resolution of the high-probability low-value regions and smoothing the impact of extreme values, the quantile normalization effectively addresses the scale discrepancy, thereby providing a more balanced and robust dataset structure for the subsequent deep learning model training.

3.2 *Model specification and training*

We implement a BNN based on a Multi-Layer Perceptron architecture, which leverages MC Dropout to approximate model uncertainty. This network is

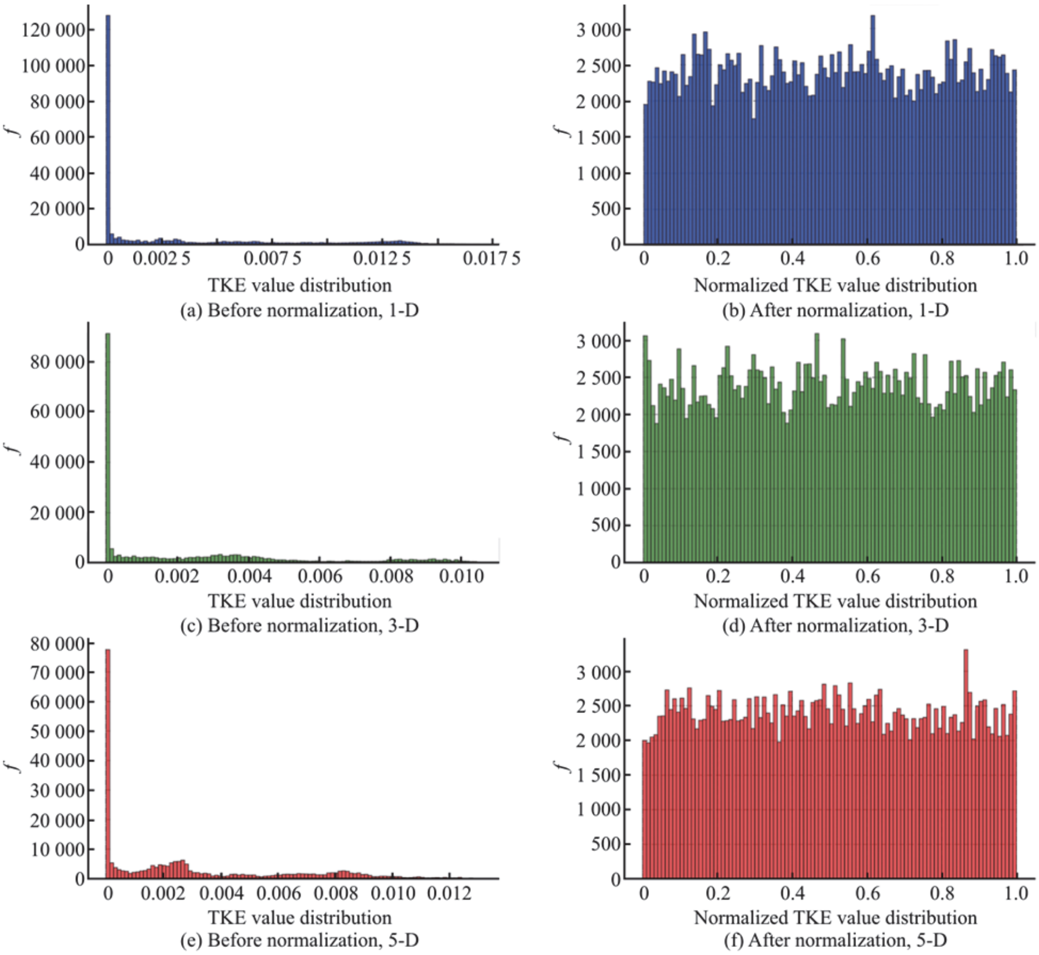


Fig. 6 Data distribution before (left-column figures) and after (right-column figures) quantile normalization

designed to generate both predictive mean estimates and corresponding uncertainty measures, rendering it well-suited for regression tasks where quantifying prediction confidence is indispensable. The model architecture comprises an input layer, a single shared hidden layer, and an output layer. The input dimension is determined by the number of features (columns), which is mapped to 200 hidden units. This is followed by a rectified linear unit (ReLU) activation function and a dropout layer with a default dropout rate of 0.2. To enhance computational efficiency and facilitate gradient flow, the network incorporates 10 residual blocks that reuse the same hidden layer. Each block applies a linear transformation, a ReLU activation, and dropout, followed by a residual connection that adds the block's input to the transformed output–this design improves training stability and mitigates the vanishing gradient problem. The output layer projects the 200-dimensional hidden representation to a flattened vector of size M_{size}, which is then reshaped into a tensor with dimensions (8, M_{dim}, M_{dim}). A Softplus activation function is applied to the final output to ensure non-negative values, a critical feature for regression tasks involving non-negative targets (e.g. TKE prediction in this study). During training, the model is optimized using the Huber loss function, which confers robustness to outliers. The Adamax optimizer is employed with an initial learning rate of 10^{-4}, paired with a StepLR learning rate scheduler that reduces the learning rate by a factor of 0.5 every 5 000 training steps.

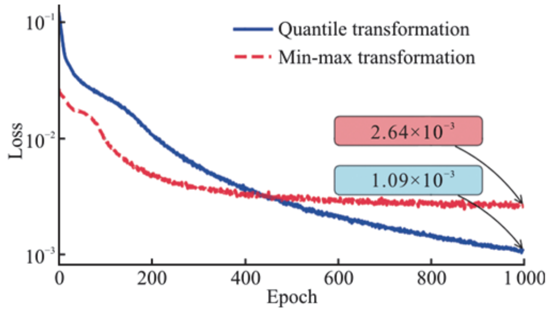


Fig. 7 Loss history of neural networks (NNs) in the training process

A key feature of this implementation is the use of MC dropout for uncertainty quantification. Unlike conventional dropout (which is only active during

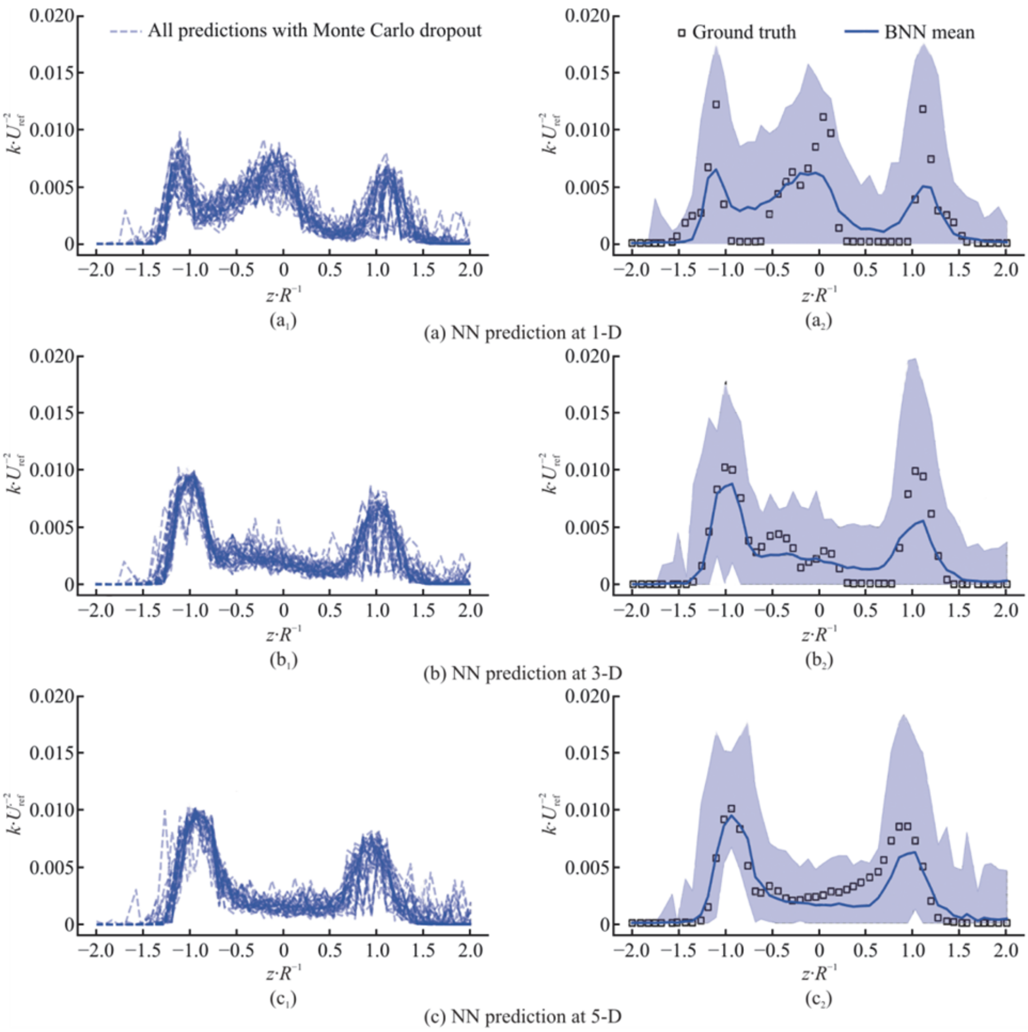


Fig. 8 NN predictions with Min-Max scaler in training set. Left figures: all predictions with MC dropout, Right figures: mean prediction with uncertainty

training), MC dropout remains enabled during the inference phase. The MC predict method performs multiple forward passes (default: 100 passes) with dropout activated, generating a distribution of predictions for each input sample. The predictive mean is calculated as the average of these samples, while the standard deviation serves as a metric for epistemic uncertainty. This approach provides a practical and scalable approximation of Bayesian inference, enabling the model to output confidence levels alongside point predictions without incurring substantial computational overhead. As such, the model establishes a principled framework for uncertainty-aware regression, with applicability in domains such as medical diagnosis, autonomous systems, and financial modelling–where reliable uncertainty estimates are pivotal for informed decision-making.

Training was conducted on an NVIDIA RTX 4060 (8GB) GPU, with each training epoch requiring approximately 2 s.

Figure 7 illustrates the Huber loss history during the training and validation processes, where the smooth decline and close alignment of both curves demonstrate thorough model convergence and excellent generalizability without any signs of overfitting or underfitting.

3.3 *Results and discussion*

This section presents a comprehensive evaluation of the proposed Bayesian deep learning framework integrated with quantile transformation for predicting TKE in wind turbine wakes. The framework's performance is rigorously assessed by comparing predictions from models trained with Min-Max scaling (data scaled to $[0,1]$ without changing the distribution) and quantile scaling against the high-fidelity CFD dataset, with a dual focus on point prediction accuracy and uncertainty quantification performance.

As it is shown in Figs. 8, 9, the Min-Max scaling approach consistently fails to capture the high-TKE

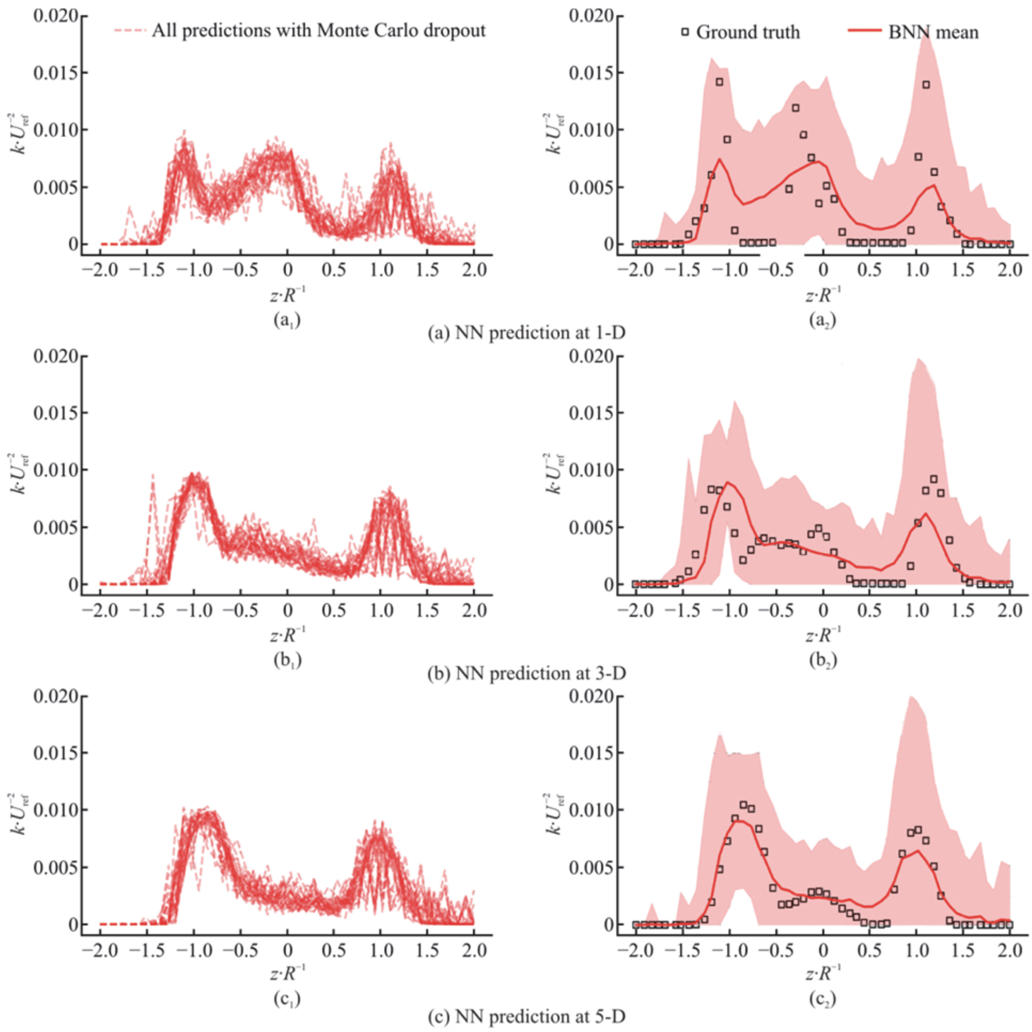


Fig. 9 NN predictions with Min-Max scaler in testing set. Left figures: all predictions with MC dropout, Right figures: mean prediction with uncertainty

regions that are critical to wake dynamics–such as those near the blade tips and the hub–resulting in predictions that cluster around low values across both the training and testing sets. Crucially, the uncertainty estimates generated by this method remain uniformly high across the entire computational domain, indicating that the model is unable to distinguish between reliable and unreliable predictions. This pervasive, uninformative uncertainty renders the uncertainty quantification practically meaningless: It provides no actionable guidance on where the model's predictions are trustworthy or where errors are likely to arise, thereby undermining the model's utility for decision-making in wind farm operations.

In contrast, the Quantile scaling method effectively reproduces the distinctive high-TKE features at all downstream locations, thus accurately capturing the physical structure of the wake–including the typical double-peak pattern and wake recovery dynamics. The uncertainty quantification it yields is notably more informative and physically meaningful: It exhibits reduced uncertainty in regions where predictions are in good agreement with high-fidelity CFD data (e.g. near the blade tips and hub) and elevated uncertainty at the wake edges where prediction confidence is naturally lower. Such a nuanced uncertainty distribution directly aligns with the model's expected reliability, providing engineers with a clear, actionable insight into where predictions are most trustworthy and where additional scrutiny may be warranted. The results are shown in Figs. 10, 11.

Overall, the comparative analysis underscores that quantile scaling fundamentally reshapes the model's uncertainty behavior–shifting from the indiscriminate, uninformative uncertainty profile of Min-Max scaling to a physically consistent and context-aware representation. This improvement ensures uncertainty estimates are no longer a mere byproduct of the model but a reliable indicator of prediction quality, which is critical for wake modeling applications. Here, iden-

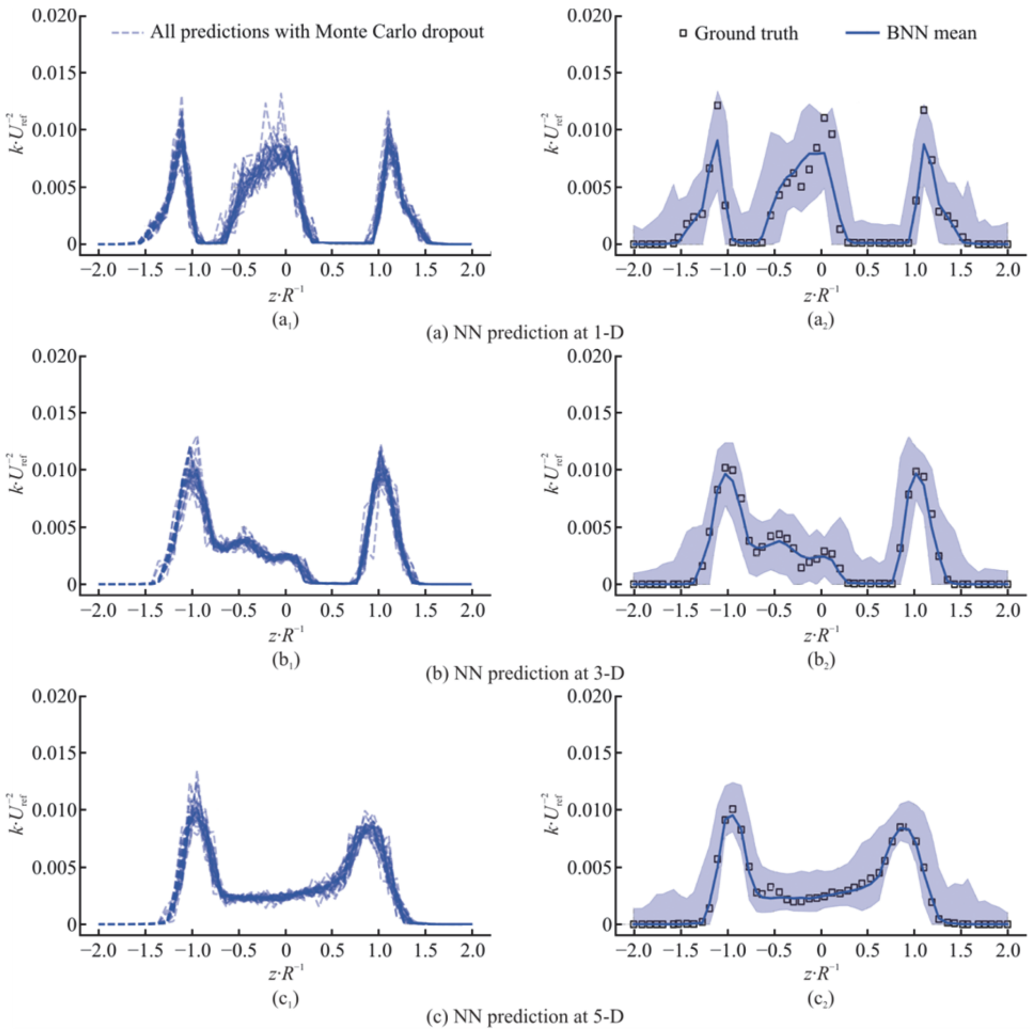


Fig. 10 NN predictions with quantile scaler in training set. Left figures: all predictions with MC dropout, Right figures: mean prediction with uncertainty

tifying low-confidence regions can directly inform wind farm turbine placement strategies or operational adjustments to optimize performance. The quantile approach thus delivers dual value: more accurate point predictions and a robust framework for uncertainty-aware decision-making. In contrast, Min-Max scaling's one-size-fits-all uncertainty fails to distinguish reliable from unreliable predictions, substantially undermining the model's practical utility.

4. Conclusions

This work addresses the fundamental challenge of modeling TKE in wind turbine wakes by proposing quantile transformation as a targeted preprocessing strategy for neural network-based prediction. The approach directly tackles the inherently highly skewed TKE distribution–where the vast majority of values cluster at low magnitudes, whereas critical high-TKE regions (concentrated near blade tips and the hub) are sparse. By nonlinearly transforming the data into a more uniform distribution, quantile transformation enables the neural network to learn equitably from the entire range of TKE values, avoiding bias toward the dominant low-TKE regions.

The key finding underscores that Quantile Transformation empowers the neural network–when paired with the BNN framework–to capture the physically meaningful high-TKE features critical for accurate wake prediction. Unlike conventional Min-Max scaling, which compresses skewed TKE data and obscures these sparse yet influential regions, Quantile Transformation preserves the relative relationships across the entire TKE spectrum. This ensures the model does not merely prioritize the dominant low-TKE values but learns the full complexity of wake dynamics, including the concentrated high-TKE zones near blade tips and the hub. The outcome is twofold: improved point prediction accuracy for TKE distributions and, more critically, physically coherent uncertainty quantification that aligns with the underl-

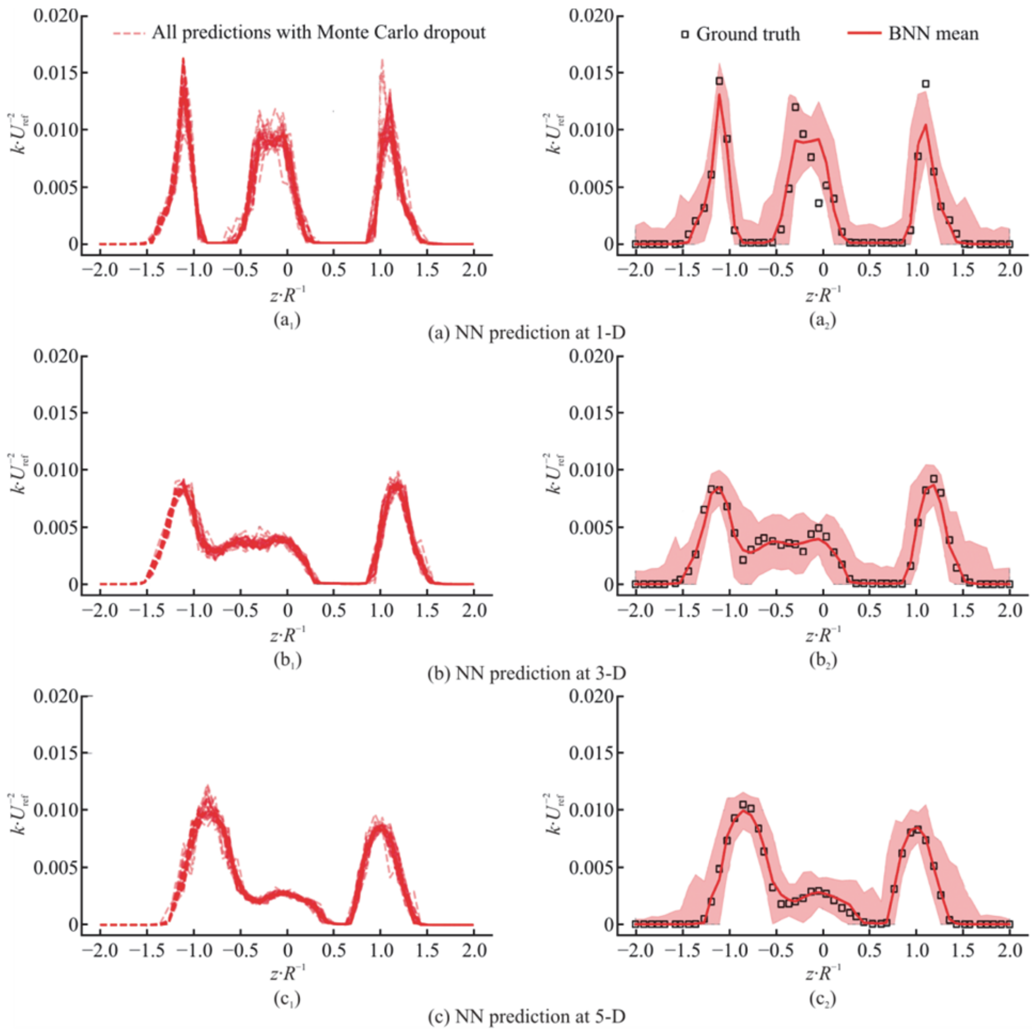


Fig. 11 NN predictions with quantile scaler in testing set. Left figures: all predictions with MC dropout, Right figures: mean prediction with uncertainty

ying physics of wake dynamics. Specifically, the model exhibits reduced uncertainty in regions where predictions are consistent with high-fidelity CFD data (e.g. expected high-TKE zones near blade tips) and elevated uncertainty at wake edges or transitional flow regions–delivering actionable insights for engineers to assess prediction reliability. This stands in stark contrast to Min-Max scaling, which produces uniform, uninformative uncertainty that provides no guidance for real-world applications.

Future research will focus on three potential directions: First, integrating quantile transformation with real-time operational data (e.g. on-site sensor measurements) to enhance the model's adaptability to dynamic wind farm conditions; second, coupling this preprocessing strategy with physics-informed neural networks to further constrain predictions to fundamental fluid dynamics principles; and third, scaling the framework to optimize wind farm layout design and real-time operational adjustments (e.g. yaw angle tuning, power output regulation) by leveraging the model's physically coherent uncertainty estimates. These extensions hold the potential to drive more efficient and reliable wind energy systems, rooted in a deeper understanding of wake dynamics and quantified uncertainty.

Conflict of interest

All authors declare that there are no other competing interests.

References

[1] Jensen N O. A note on wind generator interaction [R]. Roskilde, Denmark: Riso National Laboratory, 1983: Riso-M-2411.

[2] Katic I, Højstrup J, Jensen N O. A simple model for cluster efficiency [C]. *Proceedings of the European Wind Energy Conference (EWEC'86)*, Rome, Italy, 1987: 407-410.

[3] Frandsen S, Barthelmie R, Pryor S, et al. Analytical modelling of wind speed deficit in large offshore wind farms [J]. *Wind Energy*, 2006, 9: 39-53.

[4] Li R, Zhang J, Zhao X. Dynamic wind farm wake modelling based on a bilateral convolutional neural network and high-fidelity LES data [J]. *Energy*, 2022, 258: 124845.

[5] Li S, Zhang M, Piggott M D. End-to-end wind turbine wake modelling with deep graph representation learning [J]. *Applied Energy*, 2023, 339: 120928.

[6] Yang S, Deng X, Ti Z, et al. Cooperative yaw control of wind farm using a double-layer machine learning framework [J]. *Renewable Energy*, 2022, 193: 519-537.

[7] Vermeer L J, Sørensen J N, Crespo A. Wind turbine wake aerodynamics [J]. *Progress in Aerospace Sciences*, 2003, 39: 467-510.

[8] Stevens R J A M, Meneveau C. Flow structure and turbulence in wind farms [J]. *Annual Review of Fluid Mechanics*, 2017, 49: 311-339.

[9] Porté-Agel F, Bastankhah M, Shamsoddin S. Wind-turbine and wind-farm flows: A review [J]. *Boundary-Layer Meteorology*, 2020, 174: 1-59.

[10] Sørensen J N, Shen W Z. Numerical modeling of wind turbine wakes [J]. *Journal of Fluids Engineering*, 2002, 124(2): 393-399.

[11] Churchfield M J, Lee S, Michalakes J, et al. A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics [J]. *Journal of Turbulence*, 2012, 13: 1-32.

[12] Yang X, Sotiropoulos F, Conzemius R J, et al. Large-eddy simulation of turbulent flow past wind turbines/farms: The Virtual Wind Simulator (VWiS) [J]. *Wind Energy*, 2015, 18: 2025-2045.

[13] Ye M, Li M, Liu M, et al. Overview of data-driven models for wind turbine wake flows [J]. *Journal of Marine Science and Application*, 2025, 24: 1-20.

[14] Brunton S L, Noack B R, Koumoutsakos P. Machine learning for fluid mechanics [J]. *Annual Review of Fluid Mechanics*, 2020, 52: 477-508.

[15] Ti Z, Deng X, Yang H. Wake modeling of wind turbines using machine learning [J]. *Applied Energy*, 2020, 257: 114025.

[16] Ti Z, Deng X, Zhang M. Artificial neural networks based wake model for power prediction of wind farm [J]. *Renewable Energy*, 2021, 172: 618-631.

[17] Liu L, Liu J, Ye Y, et al. Ultra-short-term wind power forecasting based on deep Bayesian model with uncertainty [J]. *Renewable Energy*, 2023, 205: 598-607.

[18] Paladaechanan P, Zhong Y, Ye M, et al. Ensemble adaptive gated multi-fidelity neural network for Bayesian optimization: Application to hydrofoil design [J]. *Ocean Engineering*, 2026, 343: 123314.

[19] Ye M, Fan G, Wan D. Enhancing turbulent channel flow super-resolution via physics-guided deep learning with global pressure correlations [J]. *Journal of Hydrodynamics*, 2025, 37(5): 1001-1010.

[20] Ye M, Yao Z, Chen H, et al. Investigating the impact of numerical uncertainties in CFD-generated training data on neural network models for wind-turbine wake predictions [J]. *Ocean Engineering*, 2026, 343: 123200.

[21] Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning [C]. *Proceedings of the 33rd International Conference on Machine Learning*, New York, USA, 2016: 1050-1059.

[22] Krogstad P Å, Eriksen P E. "Blind test" calculations of the performance and wake development for a model wind turbine [J]. *Renewable Energy*, 2013, 50: 325-333.

[23] Ye M, Wang N, Chen H, et al. CFD simulations targeting the performance of the NTNU BT1 wind turbine using overset grids [J]. *Journal of Hydrodynamics*, 2023, 35(5): 954-962.

[24] Ye M, Chen H, Koop A. Verification and validation of CFD simulations of the NTNU BT1 wind turbine [J]. *Journal of Wind Engineering and Industrial Aerodynamics*, 2023, 234: 105336.