FlowFormer: Toward a Foundation Model for Full-Flow-Field Wind Farm Wake Modeling

Rui Li[a], Jincheng Zhang[a], Yubo Huang[a], Xiaowei Zhao[a,*]

[a] *Intelligent Control & Smart Energy (ICSE) Research Group, School of Engineering, University of Warwick, Coventry, CV4 7AL, UK*

Abstract

The foundation model has recently attracted significant attention due to its exceptional generalisability and outstanding adaptability. However, when it comes to data-driven wind farm wake modelling, due to the high cost of data generation and the complexity of flow characteristics, dimension reduction technology is still the mainstream pre-processing procedure to alleviate the significant challenge inherent in the task itself. Existing methods are still far from achieving a foundation model with high adaptability and excellent scalability. To fill the research, we propose the FlowFormer framework to serve as a foundation model. Specifically, we design a Transformer-based framework, FlowFormer, for flow field prediction by directly taking the LES-simulated data without introducing any dimensionality reduction operations. Moreover, a semi-supervised training strategy is designed to address the problem of over-fitting caused by dimension complexity. The overall mean absolute error of the developed FlowFormer is 6.660% compared to the freestream wind speed for multi-step iterative prediction. The results for a utility-scale farm consisting of 81 turbines demonstrate the high adaptability and excellent scalability of the proposed FlowFormer. Significantly, a qualitative experiment demonstrates that FlowFormer can handle changing-yaw conditions using only fixed-yaw training data, highlighting its excellent flexibility. The demo is available at https://github.com/warwick-icse/FlowFormer.

Keywords: Foundation model, Transformer, CFD simulation, Dynamic wake model

1. Introduction

Wind energy has shown significant potential and has seen rapid growth in recent years [1, 2], owing to its sustainable advantages over traditional fossil fuels. Typically, wind farms are an effective solution for optimising wind energy conversion and minimising overall costs. However, the wind turbine will severely suffer from the wake interactions, which are characterised by decreased wind velocity and increased turbulence intensity [3, 4, 5]. Specifically, such impact from upstream wind turbines to downstream turbines can lead to a 10-20% reduction in energy production [6] and a 5-15% increase in fatigue load [7] in large wind farms. Therefore, comprehensively understanding and accurately predicting wind turbine wakes is crucial for optimising turbine layouts and enabling intelligent control, ensuring efficient wind farm deployment and improving large-scale energy conversion efficiency [8, 9]. As a result, significant efforts have been devoted to wind farm wake modelling, ranging from low-fidelity analytical wake models to high-fidelity numerical wake models.

Analytical wake models are the most widely used methods for predicting wind turbine wake flows due to their low computational cost. Examples include the Jensen model [10, 11], Frandsen model [12], the 3D wake model [13], the 3DEG model [14], and the FOWFSim [15]. From the initial top-hat assumption to the Gaussian distribution of wake velocity deficits, analytical models have demonstrated good performance and have been applied to various real-world wind farms. However, these models often result in significant discrepancies between different wind farms due to their heavy reliance on empirical parameters [16, 17, 18], which typically need calibration through wind tunnel experiments or numerical simulations. Additionally, analytical models perform poorly when estimating near-wake velocity and turbulence characteristics, which are inherently more complex than those in the far-wake region [19, 20]. Thus, two distinct limitations of analytical models emerge, i.e. suboptimal near-wake prediction performance and limited generalisation across different wind farms. Besides, due to their static nature, these low-fidelity models are limited to optimising static quantities like mean power generation and are insufficient for addressing unsteady quantities such as power fluctuations and structural loads, particularly in control design scenarios [21, 22].

Numerical simulation methods, such as Large Eddy Simulation (LES) and Reynolds-averaged Navier-Stokes (RANS), are often used to evaluate wind turbine wake interactions and capture more accurate wake characteristics. The rotor in numerical simulations is typically modelled using the actuator disk method (ADM) [23, 24, 25] or the actuator line method (ALM) [26, 27, 28], which focuses on the development and features of the wake flow behind the turbine. For example, the LES/ALM coupled model can more accurately capture near-field instantaneous wakes and axisymmetric features by simulating tip and root vortices [29]. However, despite the ability of numerical simulations to capture accurate wake flow characteristics, the numerical models demand significant computational resources.

[*] Corresponding author

Email addresses: rui.li.8@warwick.ac.uk (Rui Li), jc.zhang15@gmail.com (Jincheng Zhang), yubo.h@outlook.com (Yubo Huang), xiaowei.zhao@warwick.ac.uk (Xiaowei Zhao)

For instance, to generate a 1000s Large Eddy Simulation (LES) of a 3 km × 3 km wind farm with 6 turbines, it would take about 60 h of distributed computation with 512 processors on High-Performance Computing (HPC) clusters [26]. To reduce the computational burden, control-oriented wake modelling has been gaining attention recently, with models like WFSim [30], the FAST.Farm [31], the curled wake model [32], and the FLORIDyn model [33].

In summary, low-fidelity analytical models for wind farm wake prediction are computationally efficient but lack detailed flow representation, whereas high-fidelity models are too computationally demanding for practical engineering applications. To bridge this gap, Machine Learning (ML), particularly Deep Learning (DL), holds great potential to achieve both accuracy and efficiency in wake predictions. By training on datasets established using numerical simulation results or observed flow data, ML/DL-based wake prediction models can capture key wake flow characteristics that are beyond the reach of analytical wake models while maintaining computational speed comparable to those models. For example, a surrogate model based on Delayed Proper Orthogonal Decomposition (d-POD) and Long Short-Term Memory (LSTM) network is proposed by [34] for predicting wind turbine wakes. A Reduced Order Modeling-based Wake Flow Estimation (ROM-WFE) framework is designed in [35], which establishes a nonlinear mapping between sensor measurements and low-dimensional Proper Orthogonal Decomposition (POD) coefficients. By integrating the physical and mathematical models into a deep neural network, a novel reconstruction method is developed in [36], enabling ultra-short-term wake flow prediction with sparse measured data. A Graph Neural Network (GNN) based framework is proposed in [37], which can directly operate on unstructured meshes and rapidly predict wake flow fields. An anti-noise wind field reconstruction method from Light Detection and Ranging (LiDAR) measurement is established in [38] using a Residual-Connected Physics-Informed Neural Network (RC-PINN). The Transformer-based model is also introduced by [39], which incorporates the transformer module into the conditional Generative Adversarial Network (cGAN). At the same time, the multi-fidelity concept is also introduced which includes the low-fidelity flow field as the input to reduce the dependence on large-scale high-fidelity data [40, 41, 42, 43, 39, 44].

Although a series of efforts have been poured into static wake modelling, research on developing machine learning models for predicting dynamic wind farm wake flows remains limited, with dimension reduction being a common technique to simplify modelling complexity. In [45], the POD-LSTM method is used to predict dynamic wake interactions in wind farms. Specifically, the original flow data from Simulator fOr Wind Farm Applications (SOWFA) [46] are down-sampled and reduced using POD, whereafter the reduced data along with inflow velocity and control parameters from previous time steps are used as input and reduced data in future time step as the reference for the LSTM model. In subsequent research [22], the POD dimension reduction step is omitted, and a novel Bilateral Convolutional Neural Network (BiCNN) is proposed to process the down-sampled flow field directly. In [47], a Physics-Guided Neural Network (PGNN) is developed for dynamic wake prediction, where the simulated flow field data are also down-scaled using linear interpolation during data preparation. Similarly, in [48], an autoencoder is used to map a high-dimensional nonlinear system to a low-dimensional linear system, based on data generated by the medium-fidelity WFSim model. For wind turbine wake flow field prediction, a reduction block is also used to reduce the original model input and produce the necessary modal coefficients in [49]. In [50], a Dynamic Wake Flow Estimation (DWFE) method combining Gaussian Process Regression (GPR), POD, and LSTM is proposed to predict wind turbine wake evolution from sparse measurement data. While the dimension reduction methods used in these models help address the curse of dimensionality and simplify modelling complexity, the loss of subtle flow details during the down-scaling procedure is inevitable.

To avoid the accuracy loss caused by the dimension reduction, this paper aims to develop an ML-based dynamic wind farm wake model based on full-flow-field data. Specifically, we propose a Transformer-based model, i.e. FlowFormer, to directly process the raw flow field instead of including any down-scaling operations, neither during data preparation nor the model training procedures. As the down-scaling technique indeed reduces the modelling complexity in exponential orders, the direct operation on the raw flow field will significantly increase the difficulty of the modelling procedure, especially for iterating long-term predictions. To resolve this issue, we develop a Semi-Supervised Training (SST) strategy to alleviate the over-fitting issue and enhance the iterate prediction performance. After training, an extensive set of experiments evaluates the developed model's performance. The comparison with high-fidelity data demonstrates that it can achieve accurate wake predictions in real-time. Particularly, the Mean Absolute Error (MAE) between the predicted results and high-fidelity data is 0.599 m/s, which is 6.660% of the freestream wind speed based on 100-step iterate prediction. More importantly, the proposed FlowFormer can remain stable even after 200 iterative predictions for a large-scale wind farm. Additionally, a quantitative experiment is conducted to illustrate the usability and stability of the proposed FlowFormer for handling large-scale wind farms directly. The experiments show that FlowFormer not only provides accurate results under changing-yaw conditions with only fixed-yaw training data, but also remains stable and robust after more than 200 iterative prediction steps. The novelty and contributions of this paper can be summarised as follows:

(1) A novel Transformer-based dynamic wind farm wake model is developed based on high-fidelity simulation data. Different from the existing ML-based dynamic wake models which all include the dimension reduction procedure during data preparation or model training procedures, this paper develops a deep learning method which directly processes the raw simulated flow fields, thereby avoiding any dimension reduction errors and retaining the detailed flow field features. The comparison between the proposed model and existing ML-based dynamic wake models is

Nomenclature

Abbreviations

ADM	Actuator Disk Method
ALM	Actuator Line Method
BiCNN	Bilateral Convolutional Neural Network
BP	Background Path
cGAN	conditional Generative Adversarial Network
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
d-POD	delayed Proper Orthogonal Decomposition
DWFE	Dynamic Wake Flow Estimation
DL	Deep Learning
FP	Foreground Path
GNN	Graph Neural Network
GPR	Gaussian Process Regression
HPC	High-Performance Computing
LiDAR	Light Detection and Ranging
LES	Large Eddy Simulation
LSTM	Long-Short Term Memory
MLP	Multilayer Perceptron
ML	Machine Learning
MHKA	Multi-Head Kernel Attention
MAE	Mean Absolute Error
MSE	Mean Squared Error
NREL	National Renewable Energy Laboratory
PGNN	Physics-Guided Neural Network
POD	Proper Orthogonal Decomposition
RANS	Reynolds-Averaged Navier-Stokes
RC-PINN	Residual-Connected Physics-Informed Neural Network
ROM-WFE	Reduced Order Modelling-based Wake Flow Estimation
RNN	Recurrent Neural Network
SCRTP	Scientific Computing Research Technology Platform
SOWFA	Simulator fOr Wind Farm Applications
SST	Semi-Supervised Training

Symbols

d_i	The distributed control parameter at the i^{th} time step
d^M	The distributed control parameter of the M^{th} turbine
D	The dimension of each distributed parameter
$\mathscr{D}$	The matrix of designed input variables
h	The number of heads
M	The number of wind turbines
n	The number of prediction steps
N	The number of patches
N_t	The number of time steps in LES simulations
N_x	The number of points for each subdomain in the x dimension
N_y	The number of points for each subdomain in the y dimension
P	The window size
S	The number of simulations in the LES database
$u^0{}_i$	The inflow velocity at the i^{th} time step
$\tilde{u}_i^{x,y}$	The true value of a flow snapshot on the position (x, y) at the i^{th} time step
$\hat{u}_i^{x,y}$	The approximate value of a flow snapshot on the position (x, y) at the i^{th} time step
$\mathscr{U}$	The matrix of all the flow fields
$\hat{\mathscr{U}}$	The flow field predicted by FlowFormer
$\mathbf{x}$	The input the the ML-based wake model
α	The weight of the auxiliary loss function
ϵ	The prediction error
θ	The parameters of the FlowFormer

summarised in Table 1.

(2) The proposed FlowFormer method introduces the state-of-the-art Transformer into the dynamic wind farm wake model, which is the first foundation model for wake modelling. Compared with the existing methods, FlowFormer not only demonstrates high adaptability in different yaw conditions but also shows excellent scalability in utility-scale wind farm wake simulation. Particularly, the experiment of the proposed FlowFormer on a 9 × 9 wind farm shows that our model can still hold stability and robustness for even a 200s iterative simulation.

(3) The performance of the developed FlowFormer is evaluated through comprehensive simulation tests, which include the predictions of single turbine wakes, multiple turbine wakes, yawed wakes, as well as wake interactions within a large wind farm. The results demonstrate that the proposed model can accurately predict the unsteady wind farm wakes in real-time. Specifically, the prediction error

Table 1: The comparison of the FlowFormer with other machine learning-based dynamic wake models.

Model	Model Type	Training Data	Dimension Reduction	
			Data Preparation	Model Training
POD-LSTM [45]	RNN	High-fidelity	✓	✓
BiCNN [22]	CNN	High-fidelity	✓	×
[47]	PGNN	High-fidelity	✓	×
[48]	CNN	Mid-fidelity	×	✓
ART-LSTANet [49]	CNN + RNN	High-fidelity	×	✓
DWFE [50]	GPR + LSTM	High-fidelity	×	✓
FlowFormer	Transformer	High-fidelity	×	×

on the test set is only 0.599 m/s, which is just 6.660% of the freestream wind magnitude.

(4) A quantitative experiment demonstrates the usability and stability of FlowFormer for large-scale wind farms. The model was tested under changing-yaw conditions despite being trained solely on fixed-yaw data and maintained stability for more than 2000s.

The remaining part of this paper is organized as follows: the proposed FlowFormer framework and the high-fidelity data are described in Section 2. Thereafter, the experimental results of the model are given in Section 3. The conclusions are drawn in Section 4.

2. Methodology

2.1. Problem formalization

An example wind farm is depicted in Fig. 1, consisting of M wind turbines with corresponding distributed control parameters (yaw angles in our simulations), denoted as $[d^1, d^2, \ldots, d^M]$, situated within a rectangular flow domain. For a wind farm containing M turbines, the set of input variables is represented as $\mathscr{D}$, with the shape $[M, N_t, D]$, where N_t represents the total number of time steps and D is the dimension of each distributed control parameter. Correspondingly, the shape of the output $\mathscr{U}$ is $[M, N_t, N_x \times N_y]$, where N_x and N_y denote the total number of grid points in the x- and y-directions, respectively, for each subdomain.

Additionally, when S simulations are conducted, the final shape of the input matrix and the output matrix becomes $[M \times S, N_t, D]$ and $[M \times S, N_t, N_x \times N_y]$, respectively. The high-fidelity output variables, $\mathscr{U}$, are calculated using Computational Fluid Dynamics (CFD) methods, while the prediction made by the proposed FlowFormer is denoted as $\hat{\mathscr{U}}$. The input variables for the FlowFormer include the history of flow fields $[\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_T]$, the inflow velocity $[{u^0}_1, {u^0}_2, \ldots, {u^0}_T]$, and the distributed control parameters $[d_1, d_2, \ldots, d_T]$. Thus, given an ML-based turbine wake model $f(\mathbf{x}; \theta)$, the goal is to minimise the discrepancy between the predicted flow field $\hat{\mathscr{U}}$ and the high-fidelity flow field $\mathscr{U}$ by optimising the model parameters θ:

$$\theta^* = \arg\min_{\theta} E(\theta),$$

$$E(\theta) = \frac{1}{(N_t - T) \times S} \sum_{i=T+1}^{N_t} \sum_{j=1}^{S} L(\tilde{u}_{i,j}, f(\mathbf{x}_{i,j}; \theta)), \qquad (1)$$

$$x_{i,j} = ([\tilde{u}_{i-T,j}, \tilde{u}_{i-T+1,j}, ..., \tilde{u}_{i-1,j}], [{u^0}_{i-T,j}, {u^0}_{i-T+1,j}, ..., {u^0}_{i-1,j}], [d_{i-T,j}, d_{i-T+1,j}, ..., d_{i-1,j}]),$$

where $E(\theta)$ represents the expected average error of the predicted flow field, and the loss function $L(\cdot)$ quantifies the disparity between the high-fidelity flow fields and the predicted results.

2.2. FlowFormer

The FlowFormer framework is designed based on the inherent properties of wind flow and wake interactions, which tend to be stable across adjacent time steps but are affected by external factors such as inflow velocity and distributed control parameters [22]. These external factors drive changes in the flow field, making it critical to account for both the historical data (representing past flow patterns) and external inputs (inflow velocity and control actions such as the yaw angles) in the prediction model. Therefore, our approach divides the input into two key components: the CNN-based background path and the Transformer-based foreground path, allowing for a more structured and efficient model architecture. The structure of the proposed FlowFormer is illustrated in Fig. 2.

2.2.1. Background path

The background path takes the multiple historical flow fields as the input and aims to capture the basic status for the future time step. Thus, the input and the output of the background path are highly related and similar. Therefore, we design a relatively simple CNN-based structure for the background path, consisting of multiple convolutional layers while the details of each layer can be seen in Table 2. Those layers construct an encoder-decoder structure, which can efficiently captures high-level features in the encoder and reconstructs detailed spatial information in the decoder, enabling precise pixel-level predictions.

2.2.2. Foreground path

For the foreground path, the inputs are the inflow velocity and distributed control parameters, while the output represents

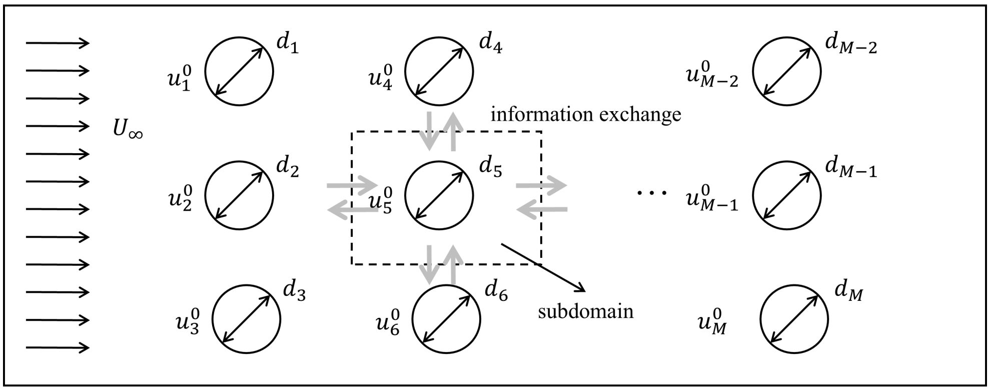


Fig. 1. A typical example of a distributed fluid system. Figure adapted from Ref. [45].

Table 2: The details of convolutional layers in the background path.

Layer	Type	Channels (In → Out)	Kernel Size	Stride	Padding
1	Conv2d	10 → 16	3×3	1	1
2	Conv2d	16 → 32	3×3	2	1
3	ConvTranspose2d	32 → 16	2×2	2	0
4	Conv2d	16 → 16	3×3	1	1
5	Conv2d	16 → 16	3×3	1	1
6	Conv2d	16 → 1	1×1	1	0

the flow field's change tendency. Because the input is low-level and low-dimensional, whereas the output is high-level and high-dimensional, this path functions as a generation task. Consequently, we use a more complex structure incorporating advanced Transformer modules.

Specifically, a Multi-layer Perceptron (MLP) layer expands the the concatenated input variables to a single-channel $(H/2) \times (W/2)$ feature map, which is then lifted to 16 channels by a 1×1 convolution. Three BaseBlocks (each = Residual Block + Transformer Block) follow. Finally, refinement layers (transpose and 1×1 convolutions) upsample and project the features before they are combined with the background path to generate the final flow prediction. The Reflection Paddings are applied for the five convolutional layer with size of (3, 1, 1, 1, 0) to keep the resolution unchanged.

Residual Block: First, the input is processed by two convolutional layers with kernel sizes (7×7) and (3×3), followed by activation using the Mish function [51]. Next, three additional convolutional layers with kernel sizes (3×3), (3×3), and (1×1) are stacked, and the output is combined with the feature maps generated from the first step through a skip connection. The resulting features are then passed to the Transformer Block.

Transformer Block: We first apply a depthwise 3×3 convolution as positional embedding to incorporate local ordering. The features are then rearranged and padded to multiples of the window size P (P is set as 14). For an input $x \in \mathbb{R}^{H \times W \times C}$, this yields $K = \frac{HW}{P^2}$ windows, each a sequence of length $T = P^2$ with channel dimension C. Within each window, we use multi-head kernel attention (8 heads and 16 channels) with L2-normalized queries/keys and a softplus feature map, which reduces the per-window attention complexity from $O(T^2)$ (dot-product) to $O(T)$. The outputs from all heads are concatenated and projected, and residual connections wrap both the attention and the MLP, i.e., $x \leftarrow x + \mathrm{Attn}(x)$ and $x \leftarrow x + \mathrm{MLP}(x)$. Thr non-local interactions are modelled within each $P \times P$ window. Finally, the windows are reversed back to the $H \times W$ layout.

2.2.3. *Loss function*

The proposed FlowFormer employs two loss functions: one for the final output ($Loss1$) and another for the foreground path ($Loss2$), both of which are defined as Mean Squared Error (MSE) functions. $Loss1$ is expressed as:

$$Loss1(\tilde{u}_{T+1}, \hat{u}_{T+1}) = \frac{1}{N_x \times N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} (\tilde{u}_{T+1}^{x,y} - \hat{u}_{T+1}^{x,y})^2 \quad (2)$$

where $\tilde{u}_{T+1}^{i,j}$ and $\hat{u}_{T+1}^{i,j}$ represent the values of the flow field at position (i, j) and time step $T+1$, obtained by LES simulation and FlowFormer, respectively. By minimising $Loss1$, the entire network is trained to learn the relationship between the input and output, enabling accurate predictions. In contrast, the reference for $Loss2$ is the difference between the flow fields at time steps T and $T + 1$, which is expressed as:

$$Loss2(\tilde{u}_{T+1}, \hat{u}_{T+1}) = \frac{1}{N_x \times N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} [(\tilde{u}_{T+1}^{x,y} - \tilde{u}_{T}^{x,y}) - (\hat{u}_{FP})_{T+1}^{x,y}]^2. \quad (3)$$

Specifically, minimising $Loss2$ encourages the foreground path to capture the variations between the flow fields at time steps T and $T + 1$, i.e., the foreground features.

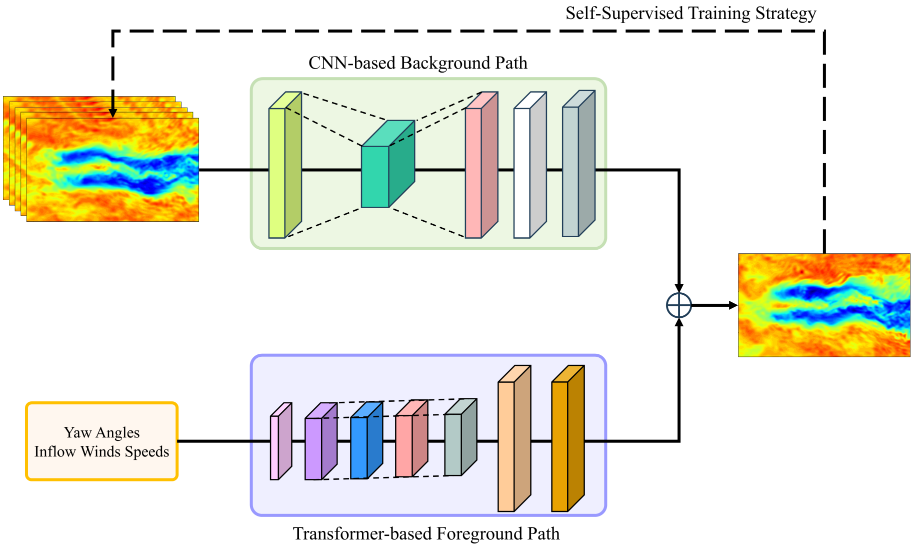

Self-Supervised Training Strategy
CNN-based Background Path
Yaw Angles
Inflow Winds Speeds
Transformer-based Foreground Path

Fig. 2. Overall architecture of FlowFormer. The model integrates two complementary paths: (i) a CNN-based background path, which captures the basic flow field evolution from historical snapshots, and (ii) a Transformer-based foreground path, which predicts wake variations driven by inflow velocity and distributed control parameters (yaw angles). The outputs of both paths are combined to produce the final flow field prediction.

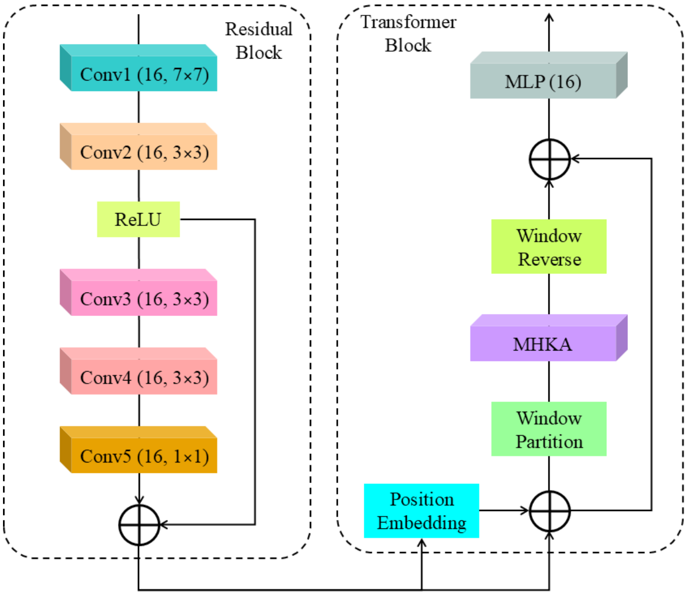


Fig. 3. The structure of BaseBlock, which is constructed by a Residual Block and a Transformer Block. The Reflection Paddings are applied for the five convolutional layer with size of (3, 1, 1, 1, 0) to keep the resolution unchanged.

In summary, the final output is optimised by $Loss1$ to closely approximate the LES data, while the output of the foreground path is constrained by $Loss2$ to learn the variation tendencies of the wake. Since the final output is the summation of the background and foreground paths, the background path is naturally driven to provide the background information. The overall loss can thus be expressed as:

$$\begin{aligned} Loss(\tilde{u}_{T+1}, \hat{u}_{T+1}) =& Loss1(\tilde{u}_{T+1}, \hat{u}_{T+1}) + \alpha Loss2(\tilde{u}_{T+1}, \hat{u}_{T+1}) \\ =& \frac{1}{N_x \times N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} \{(\tilde{u}_{T+1}^{x,y} - \hat{u}_{T+1}^{x,y})^2 \\ & + \alpha[(\tilde{u}_{T+1}^{x,y} - \tilde{u}_{T}^{x,y}) - (\hat{u}_{FP})_{T+1}^{x,y}]^2\}, \end{aligned} \tag{4}$$

where $\hat{u}_{FP}$ represents the output of the foreground path, and α controls the weight of $Loss2$. Since optimising $Loss1$ is the primary objective of the model, with $Loss2$ playing a supporting role, the value of α is set to 0.5.

2.3. Semi-supervised training

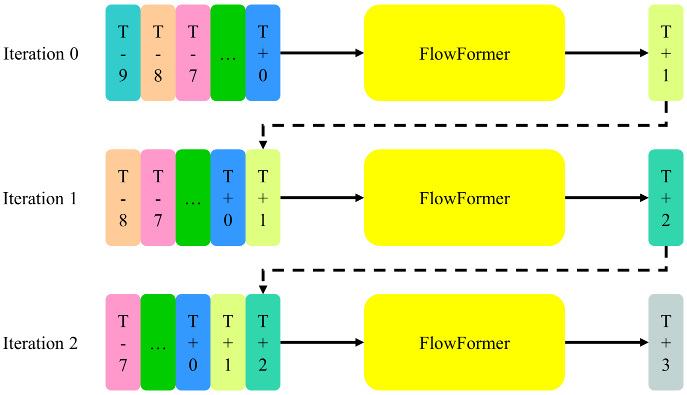


Fig. 4. Illustration of the semi-supervised training strategy in FlowFormer. The model first uses a sequence of ground-truth flow fields ($T-9 \ldots T+0$) to predict the next step ($T+1$). The predicted flow field is then fed back, together with the most recent history, as input for the next iteration, producing ($T+2$). This process is repeated for subsequent steps (here shown up to iteration 2), allowing the model to learn from its own predictions and improve stability in long-term iterative forecasting.

Table 3: The prediction errors on the test set based on the multi-step iterative prediction, while the Error shown in the last column is the percentage of the average error against the freestream wind speed.

Cases	Turbine 1 (m/s)	Turbine 2 (m/s)	Turbine 3 (m/s)	Average (m/s)	Error (%)
8 m/s	0.487	0.490	0.511	0.496	6.200
9 m/s	0.622	0.600	0.614	0.612	6.800
10 m/s	0.705	0.679	0.687	0.690	6.903

The dataset consists of 180 flow scenarios, each containing unsteady flow fields at 710 discrete time steps. For training, the first 64% of time steps from each scenario are used as the training set, the 64% to 85% time steps as the validation set, and the final 15% as the test set. The lookback time step for the flow history is set to 10. The proposed FlowFormer is trained on the training set with a batch size of 8 and a learning rate of 0.0003 using a single NVIDIA RTX5000 Ada with 32GB memory, and its performance is evaluated on the validation set after each epoch.

During training, both early-stopping and dynamic learning rate adjustment strategies are employed to mitigate over-fitting. Specifically, for early stopping, the training process is halted if the validation loss does not improve for 50 consecutive epochs. Additionally, the dynamic learning rate adjustment strategy reduces the learning rate by half if the validation loss does not decrease for 10 epochs. After training, the performance of the FlowFormer is evaluated on the test set using the mean absolute error:

$$\epsilon = \frac{1}{(N_t^{test} - T) \times S} \sum_{i=T+1}^{N_t^{test}} \sum_{j=1}^{S} |\mathscr{U}_{i,j}^{(1,1):(N_x,N_y)} - \hat{\mathscr{U}}_{i,j}^{(1,1):(N_x,N_y)}|, \quad (5)$$

where $\mathscr{U}_{i,j}^{(1,1):(N_x,N_y)}$ represents a high-fidelity snapshot of the flow field in one subdomain at time step i for the scenario indexed by j, and N_t^{test} denotes the number of time steps in the test set.

To enhance the adaptability and scalability of the proposed FlowFormer for iterative multi-step prediction, we design a semi-supervised training strategy. In this approach, the flow field predicted by the FlowFormer at future time step $T + 1$ is concatenated with the flow fields from time steps $T - 3$ to T and then fed back into the FlowFormer to predict the flow field at $T + 2$. Simultaneously, the ground truth flow field at $T + 2$ from the SOWFA simulation continues to serve as a reference to guide the network. An illustration of the first three iterations of this semi-supervised training strategy is provided in Fig. 4. Similarly, during the validation stage, the model's accuracy is assessed using iterative prediction performance instead of single-step prediction.

This strategy allows the network to become increasingly familiar with the characteristics of its own generated flow fields, thus improving its generalisability and robustness in multi-step iterative predictions, particularly when the original initialisation flow fields are no longer available during later prediction stages. In this work, we set the number of iterations to 10 for the semi-supervised training strategy, meaning that the prediction generated by the FlowFormer is reused as input up to 10 times. This ensures that the network can operate independently without relying on initialisation flow fields.

2.4. Multi-step Predictions for the Entire Wind Farm

After training, the flow field at time step $T + 1$, denoted as $\hat{u}_{T+1}$, can be predicted using the proposed FlowFormer model based on the historical flow fields $[\tilde{u}_1, \tilde{u}_2, \ldots, \tilde{u}_T]$, the inflow velocities $[u^0{}_1, u^0{}_2, \ldots, u^0{}_T]$, and the distributed control parameters $[d_1, d_2, \ldots, d_T]$. By specifying the inflow velocity and distributed control parameters at time step $T+1$, namely $u^0{}_{T+1}$ and d_{T+1}, the flow field at time step $T+2$ can be predicted using the historical data along with the previously predicted $\hat{u}_{T+1}$. In this manner, multi-step predictions can be iteratively achieved for single turbine wakes.

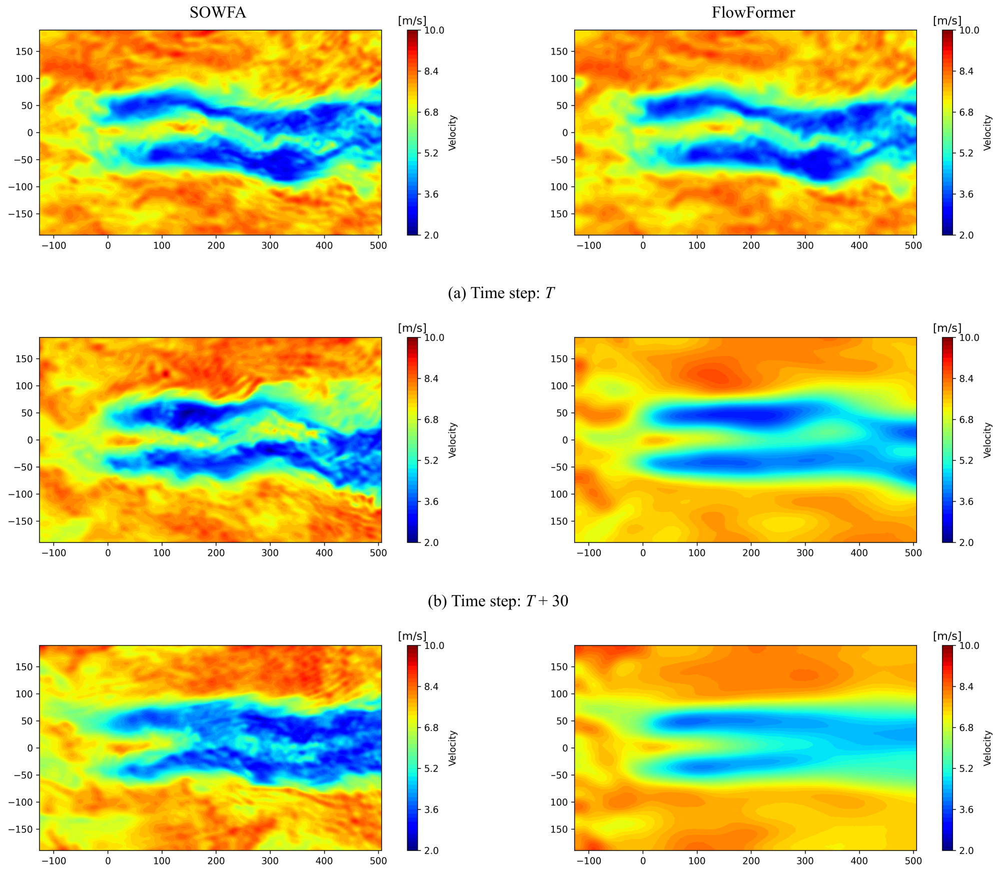
For the entire wind farm, the FlowFormer can be applied sequentially to the flow field in each subdomain from upstream to downstream. Combining the predictions from all subdomains, the overall prediction for the wind farm can be obtained. Multi-step prediction for the entire farm can then be iteratively performed in a manner similar to the single turbine case.

It is worth noting that FlowFormer can also be trained for the whole wind farm if the layout is already fixed, which is illustrated in Section 3.4.

2.5. High-fidelity data

To generate the dataset for training and testing, we use the Large Eddy Simulation (LES) solver developed by the National Renewable Energy Laboratory (NREL) for wind farm simulations, SOWFA [46], where turbine rotors are modelled as actuator lines. Simulations of three turbines [52], arranged in a row, are conducted to investigate the flow field under both freestream and upstream wake conditions. The 2D mean velocity fields at the turbine hub height around each turbine are extracted from the simulation results as the dataset. Three inflow conditions are considered, with freestream wind speeds of 8 m/s, 9 m/s, and 10 m/s. For each simulation case, 20 runs are performed, with yaw angles ranging from $[-30°, 30°]$. For each case, 1110-second simulations are carried out, with the first 400 seconds discarded to allow wake establishment, leaving 710 snapshots of the flow field per case, resulting in a dataset of flow fields $\mathscr{U}$ with dimensions $[180, 710, 126 \times 210]$. In total, about 61.2GB of original data is generated to train the FlowFormer.

The generation process was performed on a local cluster using 256 CPU cores, requiring approximately 7×10^5 CPU hours to generate the entire dataset, with each simulation taking about 46 hours. For detailed information on the simulation procedure, refer to [45].



(a) Time step: T

(b) Time step: $T + 30$

(c) Time step: $T + 100$

Fig. 5. An example case of the single-turbine wake prediction with the turbine operating in freestream condition at time steps (a) T, (b) T + 30, and (c) T + 100, where the left column is generated by SOWFA while the right column is predicted by FlowFormer. The turbine rotor is located at (0, 0) m of the 2D plane.

3. Results and discussions

To comprehensively assess the performance of the proposed FlowFormer, we begin by evaluating the prediction errors for multi-step predictions of both single-turbine wakes and multiple-turbine wakes. Next, a case study is conducted to showcase the FlowFormer's ability to capture yaw effects. Thereafter, a case study involving a simulation of a 9×9 wind farm is performed to demonstrate the scalability of the FlowFormer for utility-scale wind farms. Finally, the usability of the proposed FlowFormer for training directly on the entire wind farm is illustrated.

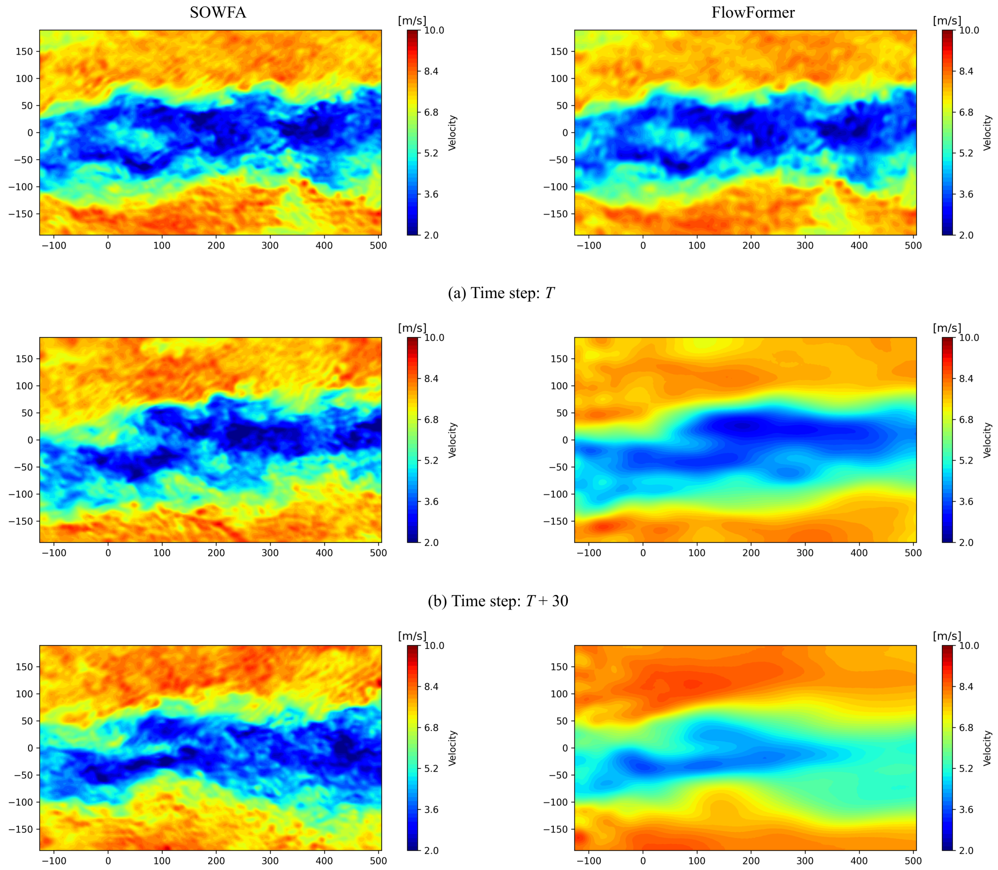
3.1. Model validations

After training, the mean absolute error between the model's predictions and the SOWFA data on the test set is 0.004 m/s, using the ground truth flow fields as initialisation input, i.e. single-step prediction. This error represents just 0.048% of the freestream wind speed, highlighting that the proposed FlowFormer can accurately capture fine details of the future flow field when accurate previous conditions are provided. To further evaluate the FlowFormer's performance with limited initialisation data, we conducted experiments using iterative prediction methods.

3.1.1. Single-turbine Wake Predictions

Based on the test set, the results of single-turbine wake predictions are presented here, where the proposed FlowFormer predicts flow fields iteratively over multiple time steps. Specifically, for the FlowFormer, the initial flow fields from time steps $T - 10$ to $T - 1$ are identical to those generated by SOWFA. The same inflow conditions and yaw angles as in SOWFA are provided from time step T to $T + n$ to predict the flow fields for the same period. In other words, only the first five ground truth flow fields are available to the network, and all subsequent flow fields are generated solely by the model.

Unlike single-step prediction, errors accumulate when wake flow fields are iteratively predicted over multiple steps, which in turn affects the accuracy of future time steps. Therefore, the prediction errors for iterative multi-step predictions provide a strong indication of the stability and long-term performance of a dynamic wind farm wake model, demonstrating its capacity for extended wake predictions. To assess the effectiveness of FlowFormer in multi-step predictions, 100 time steps are predicted iteratively for each turbine in each wind speed case, with prediction errors averaged over the time dimension. As shown in Table 3, despite the error accumulation, the overall prediction errors remain low, with an average error of 6.660% on the test set compared with freestream wind speeds.



(a) Time step: T

(b) Time step: $T + 30$

(c) Time step: $T + 100$

Fig. 6. An example case of the single-turbine wake prediction with the turbine operating in the front turbine's wake at time steps (a) T, (b) $T + 30$, and (c) $T + 100$, where the left column is generated by SOWFA while the right column is predicted by FlowFormer. The turbine rotor is located at (0, 0) m of the 2D plane.

Next, we examine the qualitative performance of the ML-based wake model. Two representative cases are chosen: a turbine operating in freestream conditions and a turbine operating in the wake of an upstream turbine. Comparisons between SOWFA and FlowFormer at time steps T, $T + 30$, and $T + 100$ are illustrated in Fig. 5 and Fig. 6, with corresponding video demonstrations available in Video 1 and Video 2.

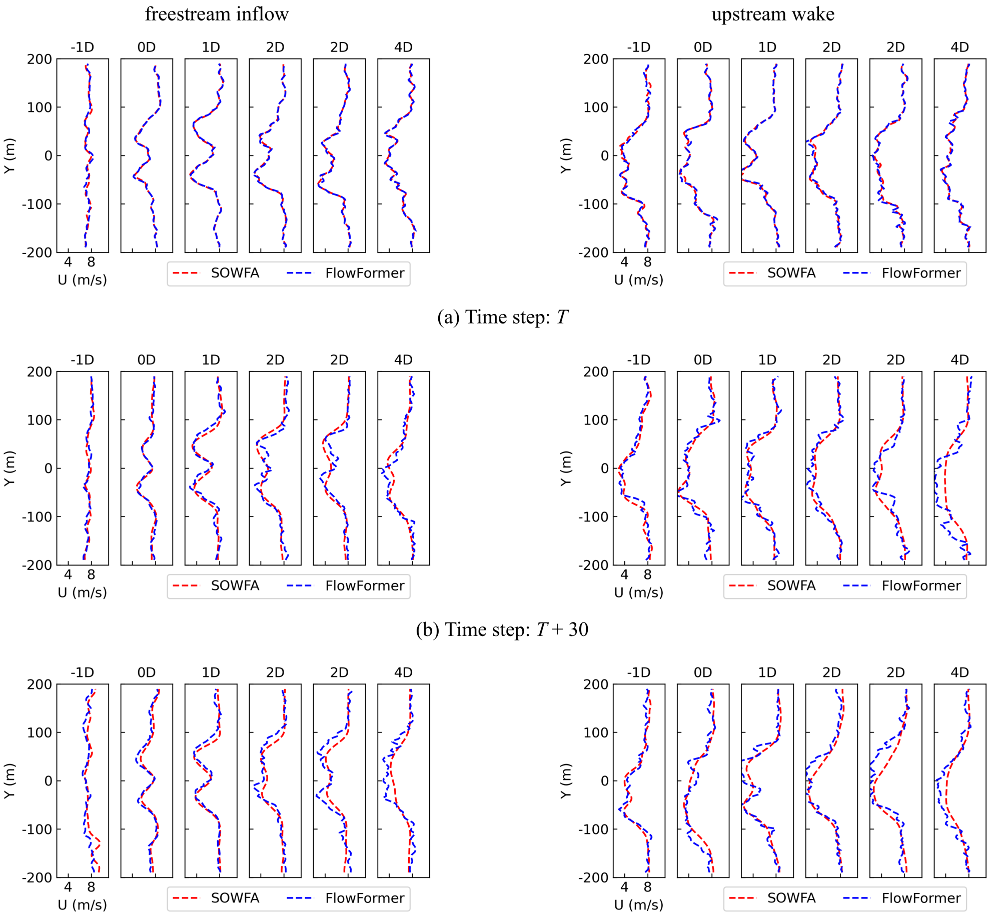
As shown in Fig. 5, the wake undergoes significant changes during the first 30 time steps, and the FlowFormer successfully captures the unsteady characteristics of the wake, maintaining accuracy throughout. Although errors accumulate over time due to the iterative nature of the predictions, the overall wake structure is well reconstructed even at time step $T + 100$. Similar results are observed when the turbine operates in the wake of an upstream turbine. In this more complex and volatile wake scenario, FlowFormer continues to accurately predict the spatial pattern and shape of the wakes even after 100 time steps, as shown in Fig. 6.

Next, the wake profiles at various streamwise locations, ranging from one rotor diameter in front of the turbine ($X = -1D$) to four rotor diameters behind the turbine ($X = 4D$), are analysed to further assess prediction performance. As shown in the left column of Fig. 7, at $X = 0D$, the freestream inflow is significantly disrupted by the turbine rotor. The wake evolves in the streamwise direction from $X = 1D$ to $X = 4D$, eventually reaching a Gaussian-like profile at $X = 4D$. In the case where the turbine operates in the wake of another turbine (right column of Fig. 7), the inflow at $X = -1D$ exhibits a near Gaussian shape induced by the upstream turbines, and the wake development from $X = 1D$ to $X = 4D$ shows similar trends as the freestream case.

From these results, it is evident that the FlowFormer predictions closely align with high-fidelity simulation results. While some finer details are lost due to error accumulation over time, the primary wake structures are accurately captured even at time step $T + 100$. In summary, the FlowFormer effectively captures the key features of unsteady turbine wakes, which is especially impressive given the inherently chaotic nature of turbulent wakes and the limited training data. Importantly, the wake predictions are achieved in real-time, requiring only 0.257 seconds of computational time per time step using a single Intel Core i7-10700 CPU.

3.1.2. *Multi-turbine wake predictions*

In this subsection, we conduct experiments on multiple-turbine wake predictions to demonstrate the ability of FlowFormer to capture wake interactions. Specifically, we consider a scenario with two turbines positioned in a row, with a down-

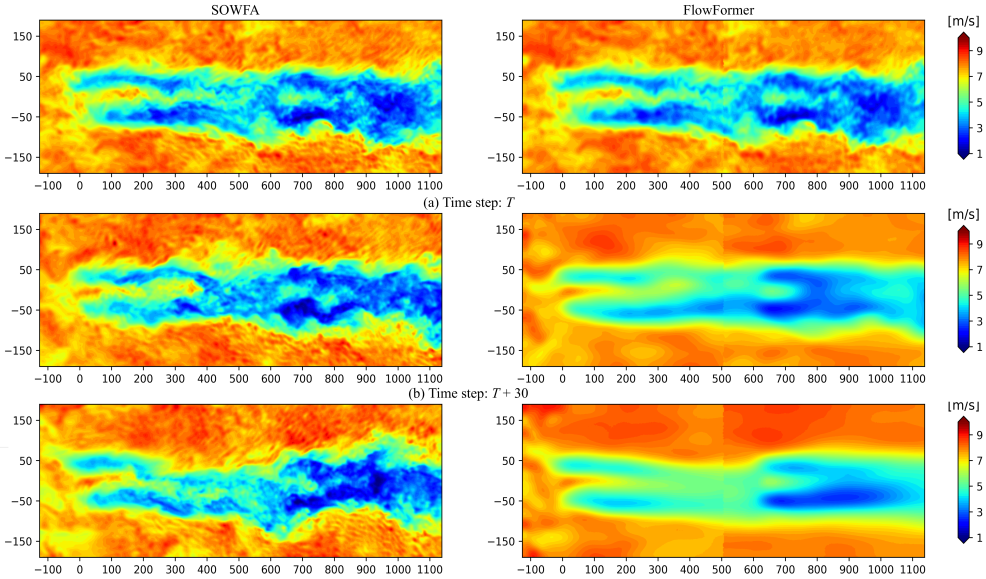


(a) Time step: T

(b) Time step: $T + 30$

(c) Time step: $T + 100$

Fig. 7. The velocity profiles of the single-turbine wake prediction with the turbine operating in freestream condition and the front turbine's wake condition at time steps (a) T, (b) T + 30, and (c) T + 100, where the red dashed line is generated by SOWFA while the blue dashed line is predicted by FlowFormer.

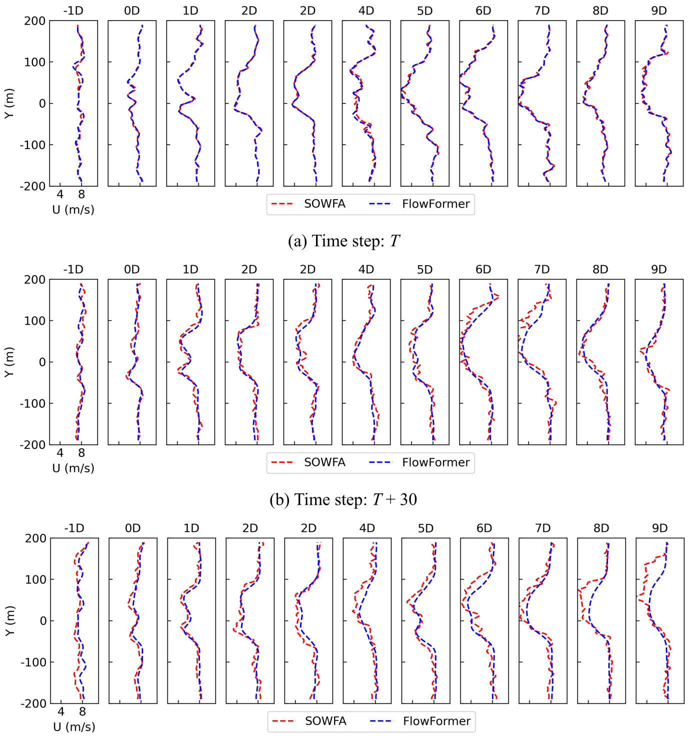


(a) Time step: T

(b) Time step: $T + 30$

(c) Time step: $T + 100$

Fig. 8. An example case of the multi-turbine wake prediction with two turbines in a row at time steps (a) T, (b) T + 30, and (c) T + 100, where the left column is generated by SOWFA while the right column is predicted by FlowFormer. The front and the rear turbine rotors are located at (0, 0) m and (632, 0) m of the 2D plane, respectively.



(a) Time step: T

(b) Time step: $T + 30$

(c) Time step: $T + 100$

Fig. 9. The velocity profiles for a two-turbine case at time steps (a) T, (b) $T+30$, and (c) $T + 100$, where the red dashed line is generated by SOWFA while the blue dashed line is predicted by FlowFormer.

stream spacing of five rotor diameters. FlowFormer is applied to iteratively predict the flow field over multiple time steps. For the front turbine, the initial inflow conditions of FlowFormer from time steps $T - 10$ to $T - 1$ are set identical to those in SOWFA, while the yaw angles for both turbines from time step T to $T + n$ are also set to match those in SOWFA. This setup allows us to evaluate FlowFormer's ability to predict the wake interactions between the two turbines as the flow field evolves over time.

An example case is shown in Fig. 8 and Video 3 to illustrate FlowFormer's ability to capture wake interactions, where the results generated by SOWFA and FlowFormer at time steps T, $T + 30$, and $T + 100$ are presented. As shown, FlowFormer's predictions align well with SOWFA simulation results, particularly during the first 30 time steps. Even after 100 time steps, the main flow features are still accurately retained. It's important to note that models without high adaptability tend to fail after just a few iterations due to error accumulation, especially in multi-turbine wake predictions, as the input values quickly deviate from the reference. The results here demonstrate FlowFormer's robust generalisation ability, successfully predicting wake interactions over iterative time steps.

To further validate the performance of the proposed FlowFormer, velocity profiles at various streamwise locations from $X = -1D$ to $X = 9D$ are shown in Fig. 9. The turbines are positioned at $X = 0D$ and $X = 5D$, where the wake deficits begin to form. Behind each turbine, key wake characteristics, such as wake deflection, recovery, and expansion, are accurately predicted by FlowFormer. Most importantly, the influence of the upstream turbine on the downstream turbine is well captured, highlighting the model's potential for large-scale wind farm predictions.

Besides, as shown in Table 4, we compare the performance of BiCNN, GNN, and the proposed FlowFormer under different inflow wind speeds with the semi-supervised training strategy. FlowFormer consistently achieves the lowest error rates across all cases, clearly outperforming the baseline models. This demonstrates FlowFormer's ability to effectively suppress error accumulation during iterative predictions, which is critical for maintaining accuracy over long time horizons. In addition, the average velocity predictions produced by FlowFormer are closer to the reference simulation values, further confirming its capability to capture essential wake dynamics. Overall, these comparative experiments highlight FlowFormer's robustness and strong generalization ability in multi-turbine wake flow prediction tasks.

3.2. *Model predictions*

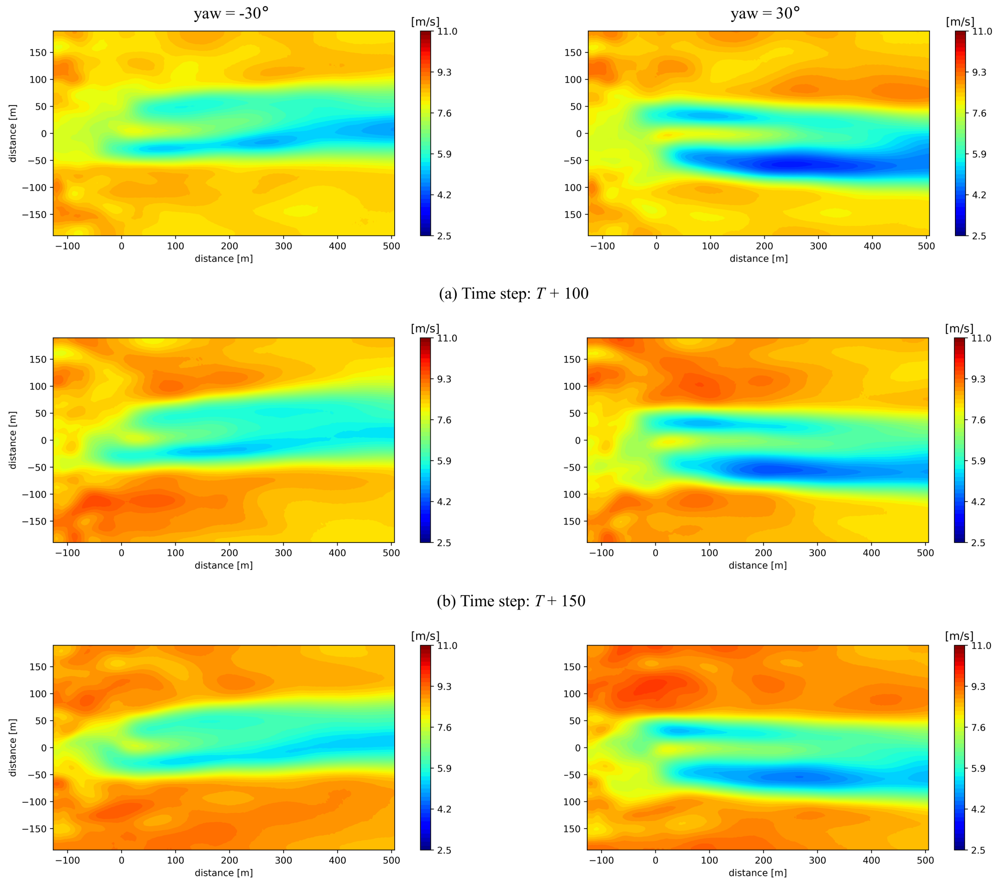
In this section, we present two extra case studies to evaluate the performance of the proposed FlowFormer model. The first case examines the yaw effects on turbine wakes by simulating a single turbine under opposite yaw angles, evaluating the model's ability to capture the flow dynamics and wake meandering induced by yaw misalignment. The second case involves a larger-scale simulation of a 81-turbine wind farm to assess the model's capability in predicting wake interactions in a utility-scale wind farm setup.

3.2.1. *The Yaw Effects on Turbine Wakes*

To demonstrate FlowFormer's ability to capture yaw effects, a case with a single turbine operating under opposite yaw angles is examined. Specifically, under identical initial conditions, two independent turbines are assigned different yaw angles ($-30°$ and $30°$). Snapshots of the flow fields at time steps $T + 100$, $T+150$, and $T+200$ are shown in Fig. 10, with a corresponding video available as Video 4. As observed, for both yaw angles, the key features of the unsteady turbine wakes are accurately captured by the model throughout the simulation. This includes the streamwise convection of flow structures in the deflected direction due to yaw, as well as the crosswind meandering of the wakes. These results demonstrate the strong performance of the proposed FlowFormer. Additionally, the model's generalisation ability is validated in this case, as the constant yaw pattern was not included during training.

3.2.2. *A 81-Turbine Test Case*

To demonstrate the capability of FlowFormer in large-scale wind farm wake predictions, we simulate two test cases each featuring a 9×9 wind turbine configuration. In both cases, the freestream wind speed is 8 m/s on average. In the first greedy case, all turbine yaw angles are held at 0°. In the second case, yaw angles vary every 10 seconds, changing randomly within the range $[-3°, 3°]$. Predicted snapshots at time steps $T + 30$, $T + 100$, and $T + 200$ are displayed in Fig. 11 and Fig. 12,



(a) Time step: $T + 100$

(b) Time step: $T + 150$

(c) Time step: $T + 200$

Fig. 10. The snapshots of the flow field around a single turbine predicted by the FlowFormer at time steps (a) $T + 100$, (b) $T + 150$, and (c) $T + 200$, where the yaw angle is $-30°$ for the left column and $30°$ for the right column. The turbine rotor is located at (0, 0) m of the 2D plane.

Table 4: Model performance comparison at different flow speeds

Model	8 m/s		9 m/s		10 m/s	
	Avg (m/s)	Error (%)	Avg (m/s)	Error (%)	Avg (m/s)	Error (%)
BiCNN	0.723	9.043	0.907	10.077	0.964	9.641
GNN	0.625	7.814	0.735	8.170	0.818	8.176
FlowFormer	0.496	6.200	0.621	6.900	0.690	6.903

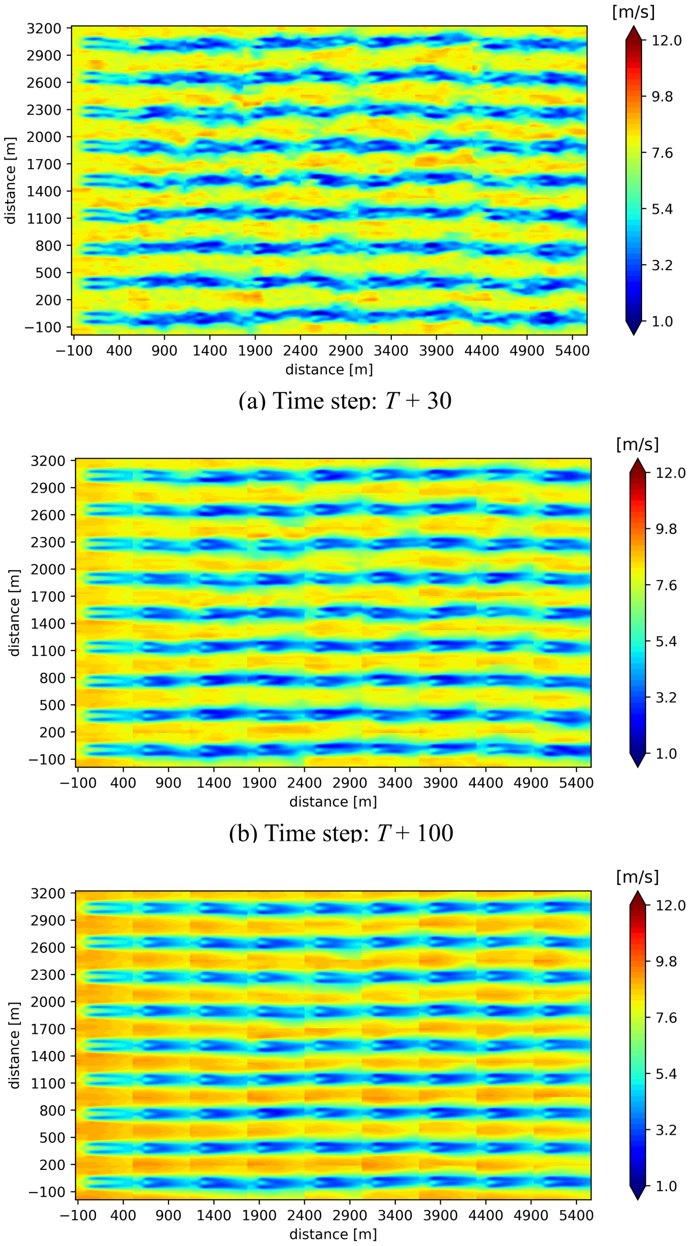
with corresponding videos available as Video 5 and Video 6. As shown, the FlowFormer effectively captures the wake interactions between turbines. The predicted unsteady flow field features, such as wake meandering and the streamwise convection of flow structures, closely resemble those produced by LES models for wind farms. This experiment highlights the model's ability to generalise to larger wind farms.

In terms of computational efficiency, the FlowFormer simulation for 200 time steps took 4159 seconds, or 20.8 seconds per time step, on a single Intel Core i7-10700 CPU. By comparison, LES models would require tens of thousands of CPU hours on an HPC cluster to simulate this scenario. This case therefore underscores the practical value of the developed model for control design in utility-scale wind farms.

3.3. *Ablation Study*

In this subsection, we conduct an ablation study to demonstrate the effectiveness of the proposed semi-supervised training strategy. Specifically, we compare the prediction performance of FlowFormer when trained with different semi-supervised iterations, i.e. 5, 10, and 15 time steps, as well as without the semi-supervised strategy. While the accuracy difference between these settings is minimal for single-step predictions, they become much more pronounced in multi-step predictions.

As shown in Table 5, without any semi-supervised training strategy, multi-step prediction errors can reach at 11.309% averagely. The performance slightly improves with 5 iterations of SST. When the iteration reaches 10, the performance is significantly enhanced. Further increases in the number of iterations lead to even better accuracy. Since GPU memory requirements increase with the number of iterations and the performance improvement between 10 and 15 iterations is relatively

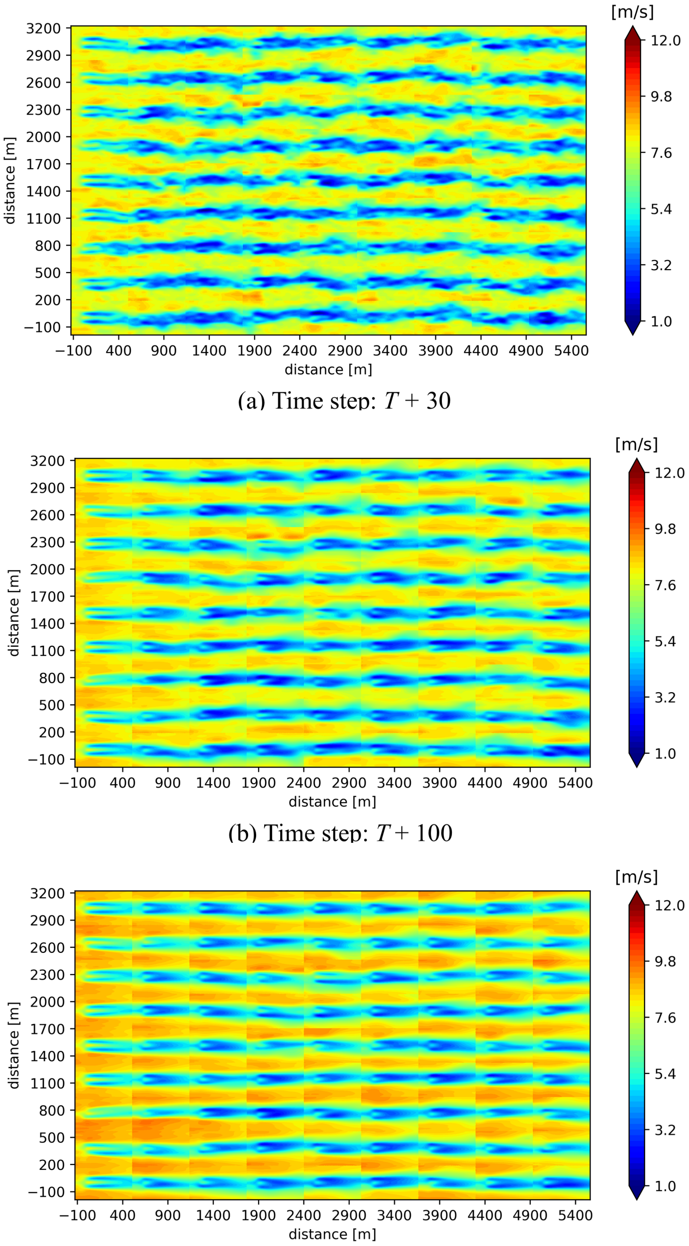


(a) Time step: $T + 30$

(b) Time step: $T + 100$

(c) Time step: $T + 200$

Fig. 11. The snapshots of the flow field around a 9 × 9 wind turbine array predicted by the FlowFormer at time steps (a) $T + 30$, (b) $T + 100$, and (c) $T + 200$, where the yaw angles of turbines are all set as 0°. The 81 turbines are located at the grid points of $[0, 632, 1264, 1896, 2528, 3160, 3792, 4424, 5056] \times [0, 379.2, 758.4, 1137.6, 1516.8, 1896.0, 2275.2, 2654.4, 3033.6]$ m of the 2D plane.



(a) Time step: $T + 30$

(b) Time step: $T + 100$

(c) Time step: $T + 200$

Fig. 12. The snapshots of the flow field around a 9 × 9 wind turbine array predicted by the FlowFormer at time steps (a) $T + 30$, (b) $T + 100$, and (c) $T + 200$, where the yaw angles change every 10 s. The 81 turbines are located at the grid points of $[0, 632, 1264, 1896, 2528, 3160, 3792, 4424, 5056] \times [0, 379.2, 758.4, 1137.6, 1516.8, 1896.0, 2275.2, 2654.4, 3033.6]$ m of the 2D plane.

non-significant, we select the model trained with 10 SST iterations for final evaluation. With more powerful GPUs and larger memory capacity, the performance of FlowFormer could be further enhanced by increasing the number of SST iterations.

3.4. Generalisability with Large-Scale Wind Farm Data

To demonstrate the strong generalisation capability of the proposed FlowFormer, we conduct experiments on a large-scale, long-duration dataset generated using an in-house GPU-accelerated simulation tool developed by our research group. Specifically, we simulate a 24-turbine layout under 15 different scenarios: three inflow wind speeds (8 m/s, 9 m/s, and 10 m/s) combined with five distinct simulations each. Each scenario runs for 3000 seconds, and the training set is sampled every 10 seconds from the raw data. During simulation, yaw angles are randomly chosen from the interval [−30°, 30°] for each turbine and remain fixed throughout the whole procedure. Although the model is only trained on these fixed-yaw scenarios, we later test it under changing yaw conditions, where yaw angles switch

Table 5: The prediction errors on the test set without SST and with different iteration steps using SST.

Settings	Iterations	Case	Turbine 1 (m/s)	Turbine 2 (m/s)	Turbine 3 (m/s)	Average (m/s)	Error (%)
No SST	-	8 m/s	0.880	0.907	0.927	0.905	11.308
		9 m/s	1.020	0.993	1.029	1.014	11.267
		10 m/s	1.137	1.121	1.146	1.135	11.347
SST	5	8 m/s	0.727	0.763	0.781	0.757	9.463
		9 m/s	0.949	0.915	0.957	0.940	10.448
		10 m/s	1.066	1.043	1.035	1.048	10.480
	10	8 m/s	0.487	0.490	0.511	0.496	6.200
		9 m/s	0.622	0.600	0.614	0.612	6.800
		10 m/s	0.705	0.679	0.687	0.690	6.903
	15	8 m/s	0.475	0.479	0.498	0.484	6.050
		9 m/s	0.601	0.585	0.598	0.595	6.607
		10 m/s	0.682	0.667	0.672	0.674	6.737

every 20 seconds.

A typical training data segment is shown in Video 7, with each flow field having dimensions of [570, 840]. As illustrated in Video 8, FlowFormer not only handles time-varying yaw angles, despite never encountering such conditions during training, but also remains stable after more than 200 iterative prediction steps (i.e., over 2000 seconds). This case study demonstrates that given sufficient data, a Transformer-based foundational model can handle scenarios not explicitly included in the training set. In future work, we plan to generate more wind farm flow field data under varying settings to further enhance FlowFormer's robustness and adaptability.

3.5. Discussion

Unlike the abundance of text and images readily available for training in other fields, obtaining CFD-based flow fields requires substantial computational resources, particularly for large-scale farms with numerous turbines. Consequently, developing a single, versatile foundational model that accommodates diverse farm layouts, turbine types, terrains and 3D flow structures remains challenging due to limited training data.

Generalisability across farm layouts. Two practical strategies can address layout variability: (i) typical turbine arrays with uniform spacing, where representative turbines are simulated and data are partitioned at the turbine level for training; and (ii) site-specific simulations, where the actual farm layout is fully simulated and then used for model training. The first approach is computationally efficient but less flexible for irregular real-world layouts, while the second provides higher accuracy for a given site but limited transferability to different configurations. In practice, selecting between the two depends on balancing computational cost, desired model generality, and site-specific complexity.

Terrain generalisability. While FlowFormer has been validated under flat-terrain scenarios, many practical sites also feature complex terrain, where topographic effects strongly influence flow characteristics. In such cases, wake evolution can be significantly altered by terrain-induced shear and turbulence. To extend FlowFormer's applicability, terrain information (e.g., elevation maps, roughness distributions) could be incorporated as additional input channels to the model. Moreover, by leveraging multi-fidelity training (e.g., combining high-fidelity LES data for flat terrain with mid-fidelity RANS data for complex terrain), the foundation model can be adapted without requiring prohibitive amounts of LES data. This direction will be pursued in our future work.

3D extension and operational deployment. This study focused on 2D hub-height fields, but FlowFormer can be naturally extended to three dimensions with two feasible pathways. (i) Multi-plane (quasi-3D) training: several horizontal planes at different heights are stacked as input channels, allowing the model to capture vertical shear, veer, and entrainment at modest computational cost; (ii) Full 3D volumetric training: the entire flow field is voxelised and processed using 3D convolution and efficient attention mechanisms (e.g., axial or kernel attention) to maintain scalability. In both cases, all three velocity components (u, v, w) can be predicted, with an additional divergence penalty to enforce approximate incompressibility. This ensures that vertical wake deflection, turbulence structures, and cross-height coherence are captured more realistically during operation. Future work will focus on implementing and validating these extensions using multi-plane and volumetric LES data.

Physics consistency. The current FlowFormer is trained entirely on LES data and does not explicitly incorporate the continuity or momentum equations. Physical consistency is therefore inherited from the high-fidelity data used during training. Future extensions could introduce physics-guided constraints into the loss function, such as introducing divergence-free regularisation to encourage mass conservation, or residual-based penalties derived from simplified Navier–Stokes equations. Such hybrid strategies would combine the generalisability of foundation models with stronger physics guarantees, particularly under long-horizon predictions or unseen conditions.

Higher-order flow statistics. While FlowFormer captures the dominant second-order characteristics of turbulence, such as overall energy distribution, purely data-driven models are inherently challenged in resolving the finest high-frequency structures. This limitation arises from the spatial smoothing effects of convolutional and downsampling operations, which can attenuate small-scale fluctuations. Future improvements could involve multi-scale architectures or physics-informed regularisations specifically designed to preserve higher-order flow features, thereby enhancing the model's fidelity across the full turbulent spectrum.

4. Conclusions

In this work, we introduced FlowFormer as a potential foundation model for wind farm wake flow field modeling. By leveraging a Transformer-based architecture and a semi-supervised training strategy, FlowFormer is trained directly on high-fidelity simulation data—totaling about 61.2 GB from SOWFA, without any dimensionality reduction steps. The model's performance was validated under a variety of inflow conditions and yaw settings, consistently achieving an overall error of only 6.660% relative to freestream wind speed in 100-second multi-step iterative predictions. A detailed analysis showed that FlowFormer accurately captures both near-wake and far-wake regions, with velocity profiles closely matching those from high-fidelity LES simulations. Furthermore, the model effectively recreates key wake structures such as yaw-induced deflection, streamwise recovery, and wake meandering, demonstrating strong adaptability to different turbine operations. Experiments on large-scale wind farms, including a 9×9 array and a 24-turbine layout, further showcased FlowFormer's scalability and robustness, even under changing yaw angles and extended prediction horizons. These results illustrate the potential of FlowFormer as a versatile, high-accuracy tool for wind farm wake modelling and provide a solid basis for future enhancements.

In the future, multi-fidelity techniques will be incorporated to reduce the reliance on time-consuming LES data. Moreover, the design concept and training strategy proposed in this paper are generic for other machine learning architectures and will be valuable for the communities to carry out further research into foundation models for wake modelling.

Declaration of Competing Interest

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

CRediT authorship contribution statement

Rui Li: Formal analysis, Conceptualization, Investigation, Methodology, Project administration, Software, Validation, Visualization, Writing - original draft. **Jincheng Zhang**: Data curation, Project administration, Software, Visualization, Writing - review & editing. **Yubo Huang**: Data curation. **Xiaowei Zhao**: Conceptualization, Funding acquisition, Formal analysis, Investigation, Methodology, Project administration, Resources, Supervision, Writing - review & editing.

Acknowledgments

This work has received funding from the (grant number: EP/Y016297/1). The authors also acknowledge the Scientific Computing Research Technology Platform (SCRTP) at the University of Warwick for providing High-Performance Computing resources.

Data availability

Data will be made available on request.

References

[1] S. C. Pryor, R. J. Barthelmie, M. S. Bukovsky, L. R. Leung, K. Sakaguchi, Climate change impacts on wind power generation, Nature Reviews Earth & Environment 1 (12) (2020) 627–643.

[2] R. Wiser, J. Rand, J. Seel, P. Beiter, E. Baker, E. Lantz, P. Gilman, Expert elicitation survey predicts 37% to 49% declines in wind energy costs by 2050, Nature Energy 6 (5) (2021) 555–565.

[3] M. Gaumond, P.-E. Réthoré, S. Ott, A. Pena, A. Bechmann, K. S. Hansen, Evaluation of the wind direction uncertainty and its impact on wake modeling at the horns rev offshore wind farm, Wind Energy 17 (8) (2014) 1169–1178.

[4] J. K. Lundquist, K. K. DuVivier, D. Kaffine, J. M. Tomaszewski, Costs and consequences of wind turbine wake effects arising from uncoordinated wind energy development, Nature Energy 4 (1) (2019) 26–34.

[5] X. Gao, B. Li, T. Wang, H. Sun, H. Yang, Y. Li, Y. Wang, F. Zhao, Investigation and validation of 3d wake model for horizontal-axis wind turbines based on filed measurements, Applied Energy 260 (2020) 114272.

[6] R. J. Barthelmie, K. Hansen, S. T. Frandsen, O. Rathmann, J. Schepers, W. Schlez, J. Phillips, K. Rados, A. Zervos, E. Politis, et al., Modelling and measuring flow and wind turbine wakes in large wind farms offshore, Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology 12 (5) (2009) 431–444.

[7] K. Thomsen, P. Sørensen, Fatigue loads for wind turbines operating in wakes, Journal of wind engineering and industrial aerodynamics 80 (1-2) (1999) 121–136.

[8] H. Dong, X. Zhao, Reinforcement learning-based wind farm control: Toward large farm applications via automatic grouping and transfer learning, IEEE Transactions on Industrial Informatics 19 (12) (2023) 11833–11845.

[9] J. J. Thomas, N. F. Baker, P. Malisani, E. Quaeghebeur, S. Sanchez Perez-Moreno, J. Jasa, C. Bay, F. Tilli, D. Bieniek, N. Robinson, et al., A comparison of eight optimization methods applied to a wind farm layout optimization problem, Wind Energy Science 8 (5) (2023) 865–891.

[10] N. O. Jensen, A note on wind generator interaction.

[11] I. Katic, J. Højstrup, N. O. Jensen, A simple model for cluster efficiency, in: European wind energy association conference and exhibition, Vol. 1, 1986, pp. 407–410.

[12] S. Frandsen, R. Barthelmie, S. Pryor, O. Rathmann, S. Larsen, J. Højstrup, M. Thøgersen, Analytical modelling of wind speed deficit in large offshore wind farms, Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology 9 (1-2) (2006) 39–53.

[13] H. Sun, H. Yang, Study on an innovative three-dimensional wind turbine wake model, Applied energy 226 (2018) 483–493.

[14] R. He, H. Yang, H. Sun, X. Gao, A novel three-dimensional wake model based on anisotropic gaussian distribution for wind turbine wakes, Applied Energy 296 (2021) 117059.

[15] A. C. Kheirabadi, R. Nagamune, Real-time relocation of floating offshore wind turbine platforms for wind farm efficiency maximization: An assessment of feasibility and steady-state potential, Ocean Engineering 208 (2020) 107445.

[16] L. Tian, W. Zhu, W. Shen, Y. Song, N. Zhao, Prediction of multi-wake problems using an improved jensen wake model, Renewable energy 102 (2017) 457–469.

[17] C. L. Archer, A. Vasel-Be-Hagh, C. Yan, S. Wu, Y. Pan, J. F. Brodie, A. E. Maguire, Review and evaluation of wake loss models for wind energy applications, Applied Energy 226 (2018) 1187–1207.

[18] Z. Liu, J. Peng, X. Hua, Z. Zhu, Wind farm optimization considering non-uniformly distributed turbulence intensity, Sustainable Energy Technologies and Assessments 43 (2021) 100970.

[19] N. Sedaghatizadeh, M. Arjomandi, R. Kelso, B. Cazzolato, M. H. Ghayesh, Modelling of wind turbine wake using large eddy simulation, Renewable Energy 115 (2018) 1166–1176.

[20] J. K. Kaldellis, P. Triantafyllou, P. Stinis, Critical evaluation of wind turbines' analytical wake models, Renewable and Sustainable Energy Reviews 144 (2021) 110991.

[21] H. Dong, J. Zhang, X. Zhao, Intelligent wind farm control via deep reinforcement learning and high-fidelity simulations, Applied Energy 292 (2021) 116928.

[22] R. Li, J. Zhang, X. Zhao, Dynamic wind farm wake modeling based on a bilateral convolutional neural network and high-fidelity les data, Energy 258 (2022) 124845.

[23] K. Nilsson, S. Ivanell, K. S. Hansen, R. Mikkelsen, J. N. Sørensen, S.-P. Breton, D. Henningson, Large-eddy simulations of the lillgrund wind farm, Wind Energy 18 (3) (2015) 449–467.

[24] M. Calaf, C. Meneveau, J. Meyers, Large eddy simulation study of fully developed wind-turbine array boundary layers, Physics of fluids 22 (1) (2010) 015110.

[25] J. Meyers, C. Meneveau, Large eddy simulations of large wind-turbine arrays in the atmospheric boundary layer, in: 48th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, 2010, p. 827.

[26] P. M. Gebraad, F. Teeuwisse, J. Van Wingerden, P. A. Fleming, S. Ruben, J. Marden, L. Pao, Wind plant power optimization through yaw control using a parametric model for wake effects—a cfd simulation study, Wind Energy 19 (1) (2016) 95–114.

[27] H. Lu, F. Porté-Agel, Large-eddy simulation of a very large wind farm in a stable atmospheric boundary layer, Physics of Fluids 23 (6) (2011) 065101.

[28] M. J. Churchfield, S. Lee, J. Michalakes, P. J. Moriarty, A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics, Journal of turbulence (13) (2012) N14.

[29] L. A. Martínez-Tossas, M. J. Churchfield, S. Leonardi, Large eddy simulations of the flow past wind turbines: actuator line and disk modeling, Wind Energy 18 (6) (2015) 1047–1060.

[30] S. Boersma, B. Doekemeijer, M. Vali, J. Meyers, J.-W. van Wingerden, A control-oriented dynamic wind farm model: Wfsim, Wind Energy Science 3 (1) (2018) 75–95.

[31] J. M. Jonkman, J. Annoni, G. Hayman, B. Jonkman, A. Purkayastha, Development of fast. farm: A new multi-physics engineering tool for wind-farm design and analysis, in: 35th Wind Energy Symposium, 2017, p. 0454.

[32] L. A. Martínez-Tossas, J. King, E. Quon, C. J. Bay, R. Mudafort, N. Hamilton, M. F. Howland, P. A. Fleming, The curled wake model: a three-dimensional and extremely fast steady-state wake solver for wind plant flows, Wind Energy Science 6 (2) (2021) 555–570.

[33] M. Becker, B. Ritter, B. Doekemeijer, D. van der Hoek, U. Konigorski, D. Allaerts, J.-W. van Wingerden, The revised floridyn model: Implementation of heterogeneous flow and the gaussian wake, Wind Energy Science Discussions (2022) 1–25.

[34] L. Zhou, J. Wen, Z. Wang, P. Deng, H. Zhang, High-fidelity wind turbine wake velocity prediction by surrogate model based on d-pod and lstm, Energy 275 (2023) 127525.

[35] Z. Luo, L. Wang, J. Xu, Z. Wang, J. Yuan, A. C. Tan, A reduced order modeling-based machine learning approach for wind turbine wake flow estimation from sparse sensor measurements, Energy 294 (2024) 130772.

[36] S. Sun, S. Cui, T. He, Q. Yao, An integrated deep neural network framework for predicting the wake flow in the wind field, Energy 291 (2024) 130400.

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

[38] R. Tian, P. Kou, Y. Zhang, M. Mei, Z. Zhang, D. Liang, Residual-connected physics-informed neural network for anti-noise wind field reconstruction, Applied Energy 357 (2024) 122439.

[39] H. Li, Q. Yang, T. Li, Wind turbine wake prediction modelling based on transformer-mixed conditional generative adversarial network, Energy 291 (2024) 130403.

[40] R. Li, J. Zhang, X. Zhao, Multi-fidelity modeling of wind farm wakes based on a novel super-fidelity network, Energy Conversion and Management 270 (2022) 116185.

[41] S. Pawar, A. Sharma, G. Vijayakumar, C. J. Bay, S. Yellapantula, O. San, Towards multi-fidelity deep learning of wind turbine wakes, Renewable Energy 200 (2022) 867–879.

[42] S. J. Anagnostopoulos, J. Bauer, M. C. Clare, M. D. Piggott, Accelerated wind farm yaw and layout optimisation with multi-fidelity deep transfer learning wake models, Renewable Energy 218 (2023) 119293.

[43] L. Zhan, Z. Wang, Y. Chen, L. Kuang, Y. Tu, D. Zhou, Z. Han, K. Zhang, Ada2mf: Dual-adaptive multi-fidelity neural network approach and its application in wind turbine wake prediction, Engineering Applications of Artificial Intelligence 137 (2024) 109061.

[44] C. Santoni, D. Zhang, Z. Zhang, D. Samaras, F. Sotiropoulos, A. Khosronejad, Toward ultra-efficient high-fidelity predictions of wind turbine wakes: Augmenting the accuracy of engineering models with machine learning, Physics of Fluids 36 (6).

[45] J. Zhang, X. Zhao, A novel dynamic wind farm wake model based on deep learning, Applied Energy 277 (2020) 115552.

[46] M. Churchfield, S. Lee, Nwtc information portal (sowfa), National Renewable Energy Laboratory.
URL https://nwtc.nrel.gov/SOWFA

[47] B. Li, M. Ge, X. Li, Y. Liu, A physics-guided machine learning framework for real-time dynamic wake prediction of wind turbines, Physics of Fluids 36 (3).

[48] K. Chen, J. Lin, Y. Qiu, F. Liu, Y. Song, Deep learning-aided model predictive control of wind farms for agc considering the dynamic wake effect, Control Engineering Practice 116 (2021) 104925.

[49] L. Xu, G. Zhou, Z. Guo, Art-lstanet: An adaptive intelligent method for wind turbine wake analysis, Engineering Applications of Artificial Intelligence 126 (2023) 106809.

[50] Z. Luo, L. Wang, Y. Fu, J. Xu, J. Yuan, A. C. Tan, Wind turbine dynamic wake flow estimation (dwfe) from sparse data via reduced-order modeling-based machine learning approach, Renewable Energy (2024) 121552.

[51] D. Misra, Mish: A self regularized non-monotonic activation function, arXiv preprint arXiv:1908.08681.

[52] J. Jonkman, S. Butterfield, W. Musial, G. Scott, Definition of a 5-mw reference wind turbine for offshore system development, Tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States) (2009).