

# Wind farm wake modeling based on deep convolutional conditional generative adversarial network

Jincheng Zhang, Xiaowei Zhao\*

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

## ARTICLE INFO

### Article history:

Received 23 March 2021

Received in revised form

4 August 2021

Accepted 8 August 2021

Available online 12 August 2021

### Keywords:

Deep learning

Generative adversarial network (GAN)

Surrogate modeling

Wake interaction

Wind farm wake

## ABSTRACT

Modeling of wind farm wakes is of great importance for the optimal design and operation of wind farms. In this work a surrogate modeling method for parametrized fluid flows is proposed for wind farm wake modeling, based on the state-of-the-art deep learning framework i.e. deep convolutional conditional generative adversarial network. Based on the proposed method and the data generated by high-fidelity large eddy simulations, a novel wind farm wake model is developed. The developed model is first validated against high-fidelity data and the results show that it achieves accurate, efficient, and robust prediction of wind turbine wake flow, at all the streamwise locations including both near wake and far wake, for both streamwise and spanwise velocity components, and at the cases with different inflow wind profiles. Then an extensive parametric study is carried out and the results show that the model generalizes well to unknown flow scenarios. Furthermore, a case study for a wind farm is investigated by the developed model. The prediction results are then compared with high-fidelity simulations, showing that the model can predict the wind farm wake flow (including both the streamwise and spanwise velocity fields) very well.

© 2021 Elsevier Ltd. All rights reserved.

## 1. Introduction

Wind energy, in particular offshore wind energy, is under intense investment in recent years. It is one of the largest renewable energy resources and is of great importance for achieving the net-zero target. In order to reduce the overall cost of wind power, individual wind turbines are usually grouped together to form large wind farms. However, the wake interactions between wind turbines have a large impact on the wind farm's overall performance [1], such as decreasing the power generations and increasing the structural loads of the rear turbines operating in full/partial wakes. The wake interactions have been investigated in the literature based on various wake models [2] ranging from low-fidelity [3–5], medium-fidelity [6–8], to high-fidelity models [9–11].

The high-fidelity CFD models solve the Navier-Stokes (NS) equations using numerical methods with the turbine modeled as actuator disks [12–14] or actuator lines [15–17]. These models can capture the detailed 3-D dynamic turbine wake behaviors. However, it is time-consuming and expensive to run. For example, the

large eddy simulation (LES) of the three-turbine array presented in Ref. [18] needs around two days to complete using 256 CPU cores on high-performance computing clusters. The low-fidelity models, on the other hand, are based on analytical formulations and can be evaluated in real time on a standard desktop, though their performance is limited by the modeling errors (including model inadequacy and parameter uncertainty [19]). Numerous analytical models have been proposed in the literature [20–22].

In order to achieve fast and accurate wake predictions, the development of new wake models is needed. Recent works include the modeling of wakes under yaw [23], the expansion of physical wake boundary [24], the 3D wakes [25,26], and the impact of atmospheric boundary layer (ABL) flows [27]. The main features of these wake models are summarized in Table 1. However, as these models are based on low-order analytical formulations, important limitations exist.

- (1) Most of these models are derived based on Gaussian wake model, which approximates the velocity in the far wake quite well but can not capture the near wake feature.
- (2) These models usually take the mean flow velocity as the input and can not accommodate a given lateral velocity profile as the inflow condition. Thus they can not take the

\* Corresponding author.

E-mail addresses: [jincheng.zhang@warwick.ac.uk](mailto:jincheng.zhang@warwick.ac.uk) (J. Zhang), [xiaowei.zhao@warwick.ac.uk](mailto:xiaowei.zhao@warwick.ac.uk) (X. Zhao).

**Table 1**

The main features of the wake models recently developed in the literature.  $U_x$  and  $U_y$  represent the streamwise wind velocity and spanwise wind velocity, respectively.

Reference	Main contributions	Wind turbine wake features					
		Far wake	Near wake	Lateral inflow profile	$U_x$	$U_y$	Multiple turbines
[23]	Wakes under yaw	Yes	No	Uniform	Yes	No	No
[24]	Physical wake boundary	Yes	No	Uniform	Yes	No	No
[25]	3D wakes	Yes	No	Uniform	Yes	No	No
[26]	Complex terrain	Yes	No	Uniform	Yes	No	Yes
[27]	Impact of ABL flows	Yes	No	Uniform	Yes	No	No

lateral variations in the upstream wind into account in the flow field predictions.

- (3) These models focus on the prediction of the streamwise wind velocity, while the spanwise velocity is usually ignored. However, the spanwise velocity can also be significant in certain scenarios such as wind farm yaw control [17].

To address these limitations, this work focuses on the development of a novel wake model based on machine learning (ML) and high-fidelity CFD simulations. By learning from the high-fidelity CFD model through flow field data, the ML model will be able to capture the main features of wake flows that cannot be captured by the low-order analytical wake models, while being evaluated as fast as these models. Currently there are very few studies on the ML-based surrogate modeling of wind farm wakes. In Ref. [28], a surrogate modeling method for flows around distributed structures was proposed and applied to build a surrogate model for wind farm wakes. The developed model employed different techniques for dimensionality reduction and fully-connected neural networks (NN) for wake predictions. In Ref. [18], a dynamic wind farm wake model was developed based on proper orthogonal decomposition (POD) for dimensionality reduction and long short-term memory networks (LSTM) for predicting wakes at future time steps using historical flow fields. The aforementioned works all included an explicit dimensionality reduction process. In this work, to avoid the explicit dimensionality reduction errors and meanwhile mitigate the overfitting issues in predicting high-dimensional target, a novel surrogate modeling method is proposed for wind farm wake modeling, which follows the generative adversarial network (GAN) framework [29], one of the most recent advancements in the field of deep learning.

As GAN is extremely powerful in generating realistic high-dimensional content, it may offer improved performance as a new ML framework for the surrogate modeling of general fluid flows. Different from explicitly fitting the training target in the supervised ML, GAN is trained implicitly to produce realistic high-dimensional content. It consists of two networks, a generator and a discriminator. The generator is designed to generate high-dimensional content while the discriminator aims to distinguish the content generated by the generator from the true content. The generator and the discriminator are trained in an adversarial way such that the generator learns to produce more realistic content so that it can ‘fool’ the discriminator, while the discriminator learns better ways to distinguish fake from true. Since its first publication, GAN has attracted extensive attention rapidly in ML field. Its developments have led a lot of exciting successes, such as the conditional generative adversarial network (CGAN) [30], deep convolutional generative adversarial network (DC-GAN) in image representations [31], image-to-image translation with CGAN [32], Wasserstein GAN [33], and cycle-consistent GAN [34]. The applications of GAN in fluid problems are still rare. A few examples include the super-resolution problems of fluid flows [35], super-resolution turbulent flow reconstructions [36], modeling of fluid flows using DC-GAN [37], and turbulence enrichment [38]. It is

expected that more works based on GAN will emerge in the future on various fluid applications, as they share an important feature - the high dimensionality of the model output (e.g. high-resolution images vs high-fidelity flow snapshots).

The surrogate modeling method proposed in this work follows the deep convolutional conditional generative adversarial network (DC-CGAN), which can take advantage of both the deep convolutional network’s ability in image processing and the CGAN’s ability in generating high-dimensional content according to specific labels/images. In order to build a robust and flexible surrogate model for fluid systems, i) the conditional GAN framework instead of the original GAN is employed so that the generator will generate the ‘realistic’ flow field according to the corresponding flow parameters; ii) the flow parameters are embedded through a fully-connected embedding layer before concatenated with the flow field and fed to the discriminator. In this way, the flow parameters from various sources can be accommodated such as the parameters defining the governing equations and boundary conditions; iii) the noise prior and the batch normalization [39] that are usually present in GAN models are excluded, as the considered fluid systems are deterministic and the surrogate model should not rely on the batch information such as the batch mean and standard deviation.

The proposed surrogate modeling method is then applied to wind farm wake modeling. Specifically, a CFD database of wind turbine wakes is first generated [28]. Then the CGAN-based wake model is trained to take the inflow wind profiles and the yaw conditions as the model input and to predict the multi-channel flow field (i.e. both streamwise and spanwise velocity fields) as the model output. After training, the prediction results are first validated against high-fidelity simulations. The results show that the developed model can predict wind turbine wakes very accurately in real time. Specifically, the prediction errors for the streamwise and spanwise velocity fields are  $0.102 \text{ ms}^{-1}$  and  $0.045 \text{ ms}^{-1}$  respectively which are just 1.1 % and 1.0 % of the corresponding value ranges. Also, the main features of turbine wakes are predicted very well, including both the overall features (such as the wake deflection with the turbine yaw angle, the wake recovery in the streamwise direction, the wake expansion in the spanwise direction) and the detailed features (such as the fluctuations in the freestream wind, the flow acceleration on both sides of the turbines, the turbines’ blockage effects on the upstream flow, and the yaw-induced spanwise velocity in the wake regions). In addition, a comprehensive parametric study is carried out and the results show that the developed wake model generalizes well to the flow scenarios that are not present in the training dataset.

To further demonstrate the use of the developed CGAN-based wake model in wind farm applications, a case study for a wind farm is carried out. For comparison purpose, the corresponding high-fidelity simulations are also carried out using the LES flow solver SOWFA (Simulator fOr Wind Farm Applications) [40]. SOWFA is developed by National Renewable Energy Laboratory (NREL) and has been validated and applied in numerous studies [16,41,42]. The results for this case study demonstrate that the developed CGAN

model is able to achieve fast, accurate, and robust predictions of the wind farm wake flows including both the streamwise and spanwise velocity fields.

The main contributions and novelties of this work are summarized as follows:

- (1) A novel wind farm wake model is developed based on high-fidelity CFD database using deep learning. The developed model can achieve accurate and robust wake predictions in real time. Compared with previous wake models based on analytical formulations, the data-based model developed in this work i) captures both the near wake and far wake features, ii) is able to accommodate given inflow velocity profile (with lateral variations) as the input for the flow field predictions, iii) achieves the predictions of both streamwise and spanwise wind velocity components, and iv) can be applied for the flow field predictions around both single turbine and multiple turbines. The accuracy and efficiency of the developed model are fully demonstrated through the validation study against CFD data and its generalization performance is evaluated through a systematic parametric study.
- (2) To build this data-based wake model, a novel surrogate modeling method for general fluid flows is proposed following the state-of-the-art deep learning framework i.e. deep convolutional conditional generative adversarial networks. The proposed method does not need explicit dimensionality reduction while the issue of overfitting in predicting high-dimensional target is mitigated.
- (3) A case study for a wind farm is carried out using both the developed wake model and the high-fidelity model SOWFA. The results fully demonstrate that the developed model can achieve accurate and robust real-time prediction on wind farm wakes.

The remaining part of this paper is organized as follows: the proposed deep learning based surrogate modeling method is described in Section 2, where the neural network structure and generative adversarial training process are explained in detail. It is then applied to the development of a novel wind farm wake model in Section 3, where the prediction performance of the developed model is evaluated first, then a case study for a wind farm is presented including the results from both the developed model and the high-fidelity LES model. Finally the conclusions are drawn in Section 4.

## 2. DC-CGAN based surrogate modeling method

A general steady-state parametrized fluid system can be described by

$$\begin{aligned} \mathcal{P}_{\mu_p}[u] &= 0, x \in \Omega_{\mu_\omega}; \\ \mathcal{B}_{\mu_b}[u] &= 0, x \in \partial\Omega_{\mu_\omega} \end{aligned} \quad (1)$$

where  $u$  is the state of the system while the differential operator  $\mathcal{P}$  (parametrized by  $\mu_p$ ), the differential operator  $\mathcal{B}$  (parametrized by  $\mu_b$ ) and  $\Omega$  (parametrized by  $\mu_\omega$ ) represent the partial differential equations (PDEs) describing the fluid systems, the boundary conditions and the flow domain respectively. Hereby we denote the flow parameters arising from the governing equations, the domain geometry and the boundary conditions as  $\mu = [\mu_p, \mu_\omega, \mu_b]$ . Given a specific value of  $\mu$ , the flow field in the domain  $\Omega$ , hereby denoted as  $U$ , can be obtained by solving Eq. (1) numerically. However, this usually requires a lot of computational resources and is time-consuming, as the degree of freedom (DoF) of the discretized PDEs is usually very high. This section is devoted to developing a

surrogate modeling method to approximate the mapping between  $\mu$  and  $U$  so that fast and accurate predictions of  $U$  can be achieved. The proposed method is based on the state-of-the-art deep learning technique DC-CGAN. It is illustrated in Fig. 1, including the generation of training data in Fig. 1(A), the GAN structure and training in Fig. 1(B), and the online prediction in Fig. 1(C). They are detailed as below.

### 2.1. Training dataset

The surrogate model is trained based on a set of samples where each sample consists of the flow parameter  $\mu$  and the corresponding flow field  $U$ , as shown in Fig. 1(A). In order to generate the training dataset, a sampling method is usually employed to generate a set of flow parameters  $[\mu^1, \mu^2, \dots, \mu^N]$  where  $N$  represents the sample size and  $\mu^i$  represents the flow parameters arising from the governing equation, the domain geometry and the boundary conditions (i.e.  $\mu^i = [\mu_p^i, \mu_\omega^i, \mu_b^i]$ ). Then a set of CFD simulations are carried out for each flow parameter  $\mu^i$  so that the corresponding flow field  $U^i$  can be obtained, as shown in Fig. 1 (A). After data generation, all the flow parameters are collected as the training input matrix  $\mathcal{X}$  of shape  $[N, N_\mu]$  where  $N_\mu$  is the dimension of the flow parameter and each row contains a sample of the flow parameter. All the corresponding high-fidelity flow field data are collected as the training target matrix  $\mathcal{Y}$  of shape  $[N, N_1, \dots, N_d, C]$  where  $[N_1, \dots, N_d]$  is the spatial resolution of the  $d$ -dimension flow domain,  $C$  is the number of the channels of the flow field data, and each row contains a sample of the flow field in the  $d$ -dimension domain with  $C$  channels. Each channel usually represents a color such as red, green or blue of an RGB image in image processing while it is used to represent a flow quantity such as streamwise velocity or spanwise velocity in this work. We mention that the dimensions of both the training input  $\mu$  and the training target  $U$  are typically very high as the former can include the boundary conditions (such as the inflow velocity) at discrete points while the latter can include multi-channel flow field on a grid of high dimension. It is this high-dimensionality that makes the surrogate modeling of such systems very challenging.

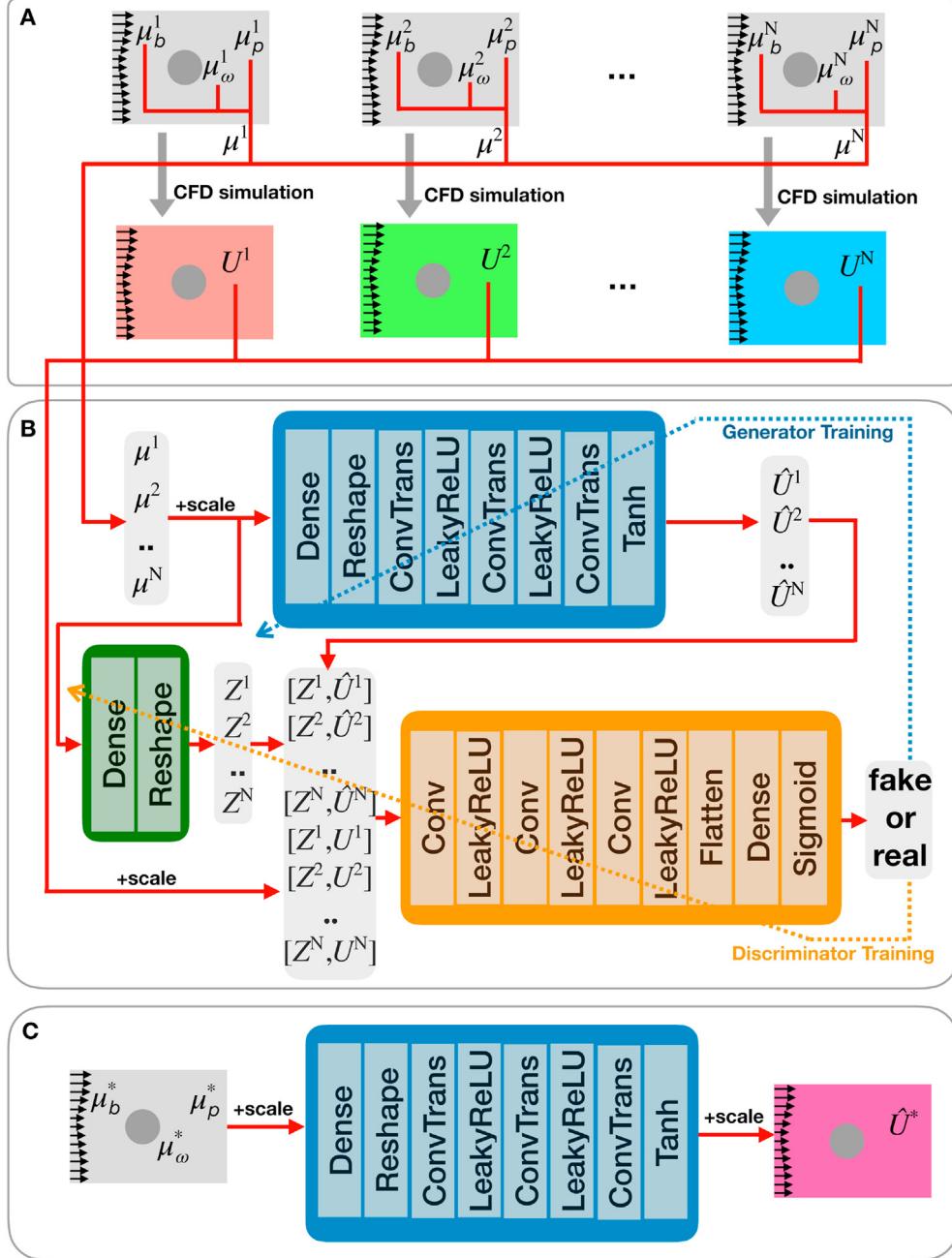
### 2.2. GAN structure

After obtaining the training dataset, the DC-CGAN based surrogate model is constructed, which is illustrated in Fig. 1 (B). It consists of a generator and a discriminator. The generator, as shown in shaded blue in Fig. 1 (B), takes the flow parameters as the input, processes it through a dense layer and a reshape layer which are then followed by a series of transposed convolution layers, and finally returns the flow field prediction  $\hat{U}$  as the output. The input-output mapping of the generator, denoted as  $\mathcal{G}$ , can be expressed as

$$\mathcal{G}(\mu) = (\sigma \circ \mathcal{T}_g)^{L_g} \circ \mathcal{R} \circ \mathcal{N}_g(\mu) \quad (2)$$

where  $\circ$  represents the function composition,  $L_g$  represents the number of transposed convolution layers, and  $\mathcal{N}$ ,  $\mathcal{R}$ ,  $\mathcal{T}$  and  $\sigma$  represent the mappings of the dense layer, the reshape layer, the transposed convolution layer, and the activation respectively. In this work, as shown in Fig. 1 (B), LeakyReLU is used for the activations in the intermediate layers and the hyperbolic tangent function is used for the last layer. The subscript  $g$  in Equation (2) indicates the corresponding mappings rely on the trainable weights which will be updated during the generator training.

The discriminator, as shown in shaded orange in Fig. 1 (B), takes the data pair of the embedded flow parameter  $Z$  and the



**Fig. 1.** The flowchart illustrating the proposed DC-CGAN based surrogate modeling method.

corresponding flow field  $U$  or  $\hat{U}$  (real or generated) as the input, processes it through a series of convolution layers, and finally returns a single classification indicator (i.e. fake or real) as the output. The main difference between CGAN and GAN is that the labels (here the flow parameters) are combined with the corresponding flow field for the examination by the discriminator, while GAN only distinguishes the generated flow field from the real flow field without the labeling information. Therefore, the CGAN structure is more suitable for the surrogate modeling of parametrized fluid flows, as the unique correspondence between the flow parameter and the flow field can be established. Here in the proposed discriminator structure, an embedding layer which consists of a dense and a reshape layer (as shown in shaded green in Fig. 1 (B)) is employed to map the flow parameters to the same shape as

the flow field so that it can be concatenated with the flow field and fed to the discriminator. The input-output mapping of the discriminator including the embedding part can be expressed as

$$\mathcal{D}([\tilde{U}, \mu]) = \sigma \circ \mathcal{N}_d \circ \mathcal{F} \circ (\sigma \circ \mathcal{C}_d)^{L_d} ([\tilde{U}, \mathcal{R} \circ \mathcal{N}_e(\mu)]) \quad (3)$$

where  $\tilde{U}$  can be the real flow field  $U$  or the generated flow field  $\hat{U}$ ,  $L_d$  represents the number of convolution layers, and  $\mathcal{F}$  and  $\mathcal{C}$  represent the mappings of the flatten layer and the convolution layer respectively. As shown in Fig. 1 (B), LeakyReLU is used for the activation in the intermediate layers while the sigmoid function is used for the last layer so that a value between 0 and 1 can be returned as the output for the binary classification. The subscripts  $d$  and  $e$  in Eq. (3) indicate that the corresponding mappings in the

discriminator and the embedding part rely on the trainable weights which will be updated during the discriminator training.

### 2.3. GAN training

**Algorithm 1.** Surrogate modeling of parametrized fluid flows based on DC-CGAN

---

```

1 % The offline training
2 Generate  $N$  samples of the flow parameters:  $[\mu^1, \mu^2, \dots, \mu^N]$ .
3 Run CFD simulations to generate the corresponding flow field data  $[U^1, U^2, \dots, U^N]$ .
4 Preprocess the flow parameters and the flow field data by MinMaxScalers.
5 Set the batch size  $N_b$ ; Set the total number of training iterations  $N_{tr}$ .
6 for  $j = 1$  to  $N_{tr}$  do
7   Sample a random batch of training data:  $\{[U^i, \mu^i], 1 \leq i \leq N_b\}$ .
8   Generate the flow field predictions  $\{\hat{U}^i, 1 \leq i \leq N_b\}$  by propagating  $\{\mu^i, 1 \leq i \leq N_b\}$  through the generator.
9   Train the discriminator to minimize  $\varepsilon_{\mathcal{D}}$  by feeding  $\{[U^i, \mu^i], 1 \leq i \leq N_b\}$  and  $\{\hat{U}^i, \mu^i, 1 \leq i \leq N_b\}$  to the discriminator.
10  Train the generator to minimize  $\varepsilon_{\mathcal{G}}$  by feeding  $\{\mu^i, 1 \leq i \leq N_b\}$  to the whole network. Only the trainable weights inside the generator are trained while the trainable weights in the discriminator (including the embedding part) are kept frozen.
11 end for
12 % The online prediction
13 Set the test flow parameter  $\mu^*$  and process it through the MinMaxScaler.
14 Predict the scaled flow field by propagating the scaled flow parameter through the generator network.
15 Obtain the final flow field prediction by scaling back the generator output.

```

---

The discriminator is trained to distinguish the pair of real flow field and flow parameter from the pair of generated flow field and flow parameter, while the generator is trained to generate realistic flow field such that the generated data pair is not distinguishable from the real data pair. Specifically, the discriminator is trained to minimize the classification error defined as

$$\varepsilon_{\mathcal{D}} = \frac{1}{N_b} \sum_{i=1}^{N_b} -\log \mathcal{D}([U^i, \mu^i]) + \frac{1}{N_b} \sum_{i=1}^{N_b} -\log(1 - \mathcal{D}([\hat{U}^i, \mu^i])) \quad (4)$$

where  $\{[U^i, \mu^i], 1 \leq i \leq N_b\}$  is a batch of training samples consisting of the real flow field and the corresponding flow parameters, and  $\{[\hat{U}^i, \mu^i], 1 \leq i \leq N_b\}$  is a batch of training samples consisting of the fake flow field generated by the generator and the corresponding flow parameters. The data batches are fed into the discriminator network to minimize  $\varepsilon_{\mathcal{D}}$ , so that the generated and real data pair can be classified as fake (i.e. 0) and real (i.e. 1) by the discriminator after training. The discriminator training is illustrated by the dashed line colored in orange in Fig. 1 (B).

The generator is trained to minimize the classification error defined as

$$\varepsilon_{\mathcal{G}} = \frac{1}{N_b} \sum_{i=1}^{N_b} -\log \mathcal{D}([\mathcal{G}(\mu^i), \mu^i]) \quad (5)$$

Here  $\{\mu^i, 1 \leq i \leq N_b\}$  is a batch of flow parameters which are fed into the generator to generate flow field that is then examined by the discriminator. The minimization of the classification error  $\varepsilon_{\mathcal{G}}$  thus guides the generator to produce data pairs which are likely to be classified as real by the discriminator. As can be seen from Eq. (5), the generator training involves the whole CGAN network including the generator and the discriminator. We mention that the trainable weights within the discriminator network are kept frozen during the training process while the trainable weights inside the

generator are trained to minimize  $\varepsilon_{\mathcal{G}}$ . The generator training is illustrated by the dashed line colored in blue in Fig. 1 (B).

The discriminator training and the generator training are carried out alternatively until the generator can produce realistic flow field that is not distinguishable from the real flow field obtained by the high-fidelity simulations. All the training data including the training input and the training target are standardized before being fed into the NN for training. The Adam optimization algorithm [43]

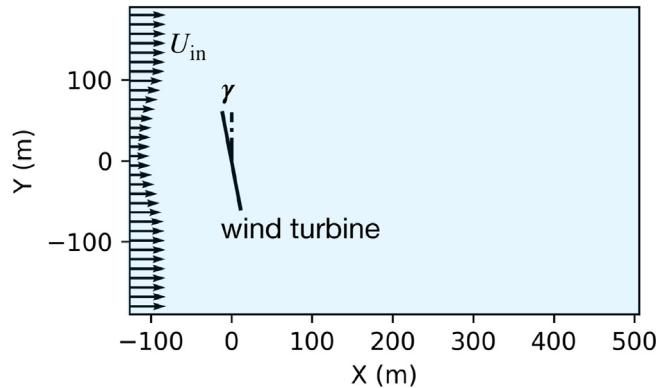
is used for the NN training and the model is implemented based on the ML package Keras [44] with Tensorflow [45] backend. The training is carried out using NVIDIA Tesla K80 GPU. After training, the generator can be used as the surrogate model of the parametrized fluid systems for the online prediction of the flow field with flow parameters as the model input, see Fig. 1 (C). The overall process including both the offline training and the online prediction is summarized as Algorithm 1.

### 3. Application to wind farm wake modeling

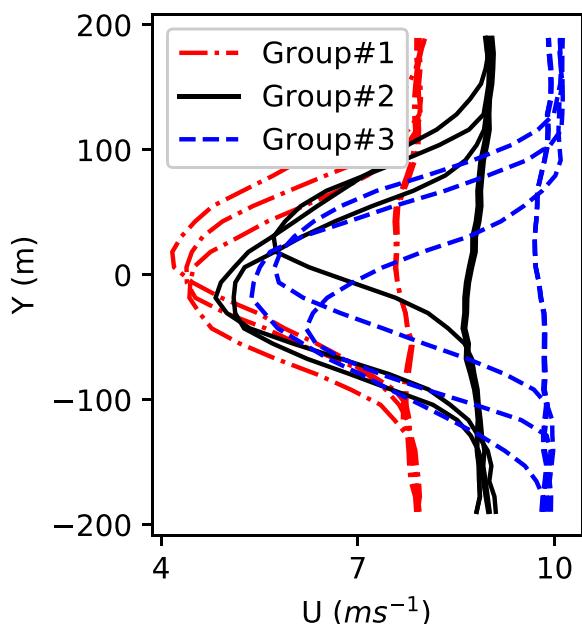
This section is devoted to the development of an accurate and efficient data-based model of wind farm wakes, by employing the surrogate modeling method proposed in Section 2. In particular, we focus on the prediction of the wind velocity field (including the streamwise and spanwise velocity fields) around the wind turbine with the inflow wind profile  $U_{in}$  and the turbine yaw angle  $\gamma$  as the model input. The flow domain and the input flow parameter are illustrated in Fig. 2. The training data, which is generated by high-fidelity CFD simulations, is described in Section 3.1. The model training and validation are then presented in detail in Section 3.2. In Section 3.3, a comparison study is carried out to demonstrate the performance of the developed CGAN model compared to the POD-NN method [28,46,47] which is based on dimensionality reduction and supervised ML approach. Next, a parametric study is carried out in Section 3.4 to systematically evaluate the generalization performance of the developed CGAN wake model. Finally a case study is investigated in Section 3.5 to fully demonstrate the use of the developed model in wind farm applications.

#### 3.1. High-fidelity data

To generate the training dataset, the high-fidelity LES flow solver SOWFA [40] is employed to solve the filtered Navier-Stokes equations where the turbine rotors are modeled as actuator lines. In this work, in order to capture the wake interactions in the training dataset, the cases of three turbines operating in a row are



**Fig. 2.** The illustration of the flow domain and the input flow parameter including the inflow wind profile  $U_{in}$  and the turbine yaw angle  $\gamma$ .



**Fig. 3.** The illustration of the inflow wind profiles for selected samples that are representatives of the training dataset.

simulated and then the 2D mean velocity field around each turbine at the turbine hub height is extracted from the simulation data. Thus each simulation can generate three training samples i.e. the flow field around the 1st, the 2nd and the 3rd turbines. In order to cover a wide range of inflow conditions, three groups of large eddy simulations are carried out where different freestream mean wind speeds at 8 ms<sup>-1</sup>, 9 ms<sup>-1</sup> and 10 ms<sup>-1</sup> are used for different groups. An illustration of the inflow wind profiles is given in Fig. 3, where Group#1, Group#2 and Group#3 correspond to the simulation groups with the freestream mean wind speed of 8 ms<sup>-1</sup>, 9 ms<sup>-1</sup> and 10 ms<sup>-1</sup> respectively. As shown in Fig. 3, for each group, the inflow wind profiles include the 'flat' profiles representing the freestream wind conditions and the 'bell-shape' profiles representing the incoming wind conditions induced by the upstream turbine wakes. Furthermore, in order to capture the yaw effects in the training dataset, 30 simulations have been carried out for each simulation group, where the turbine yaw angles are randomly sampled in the interval [-30°, 30°] for each simulation case. Therefore, in total, 90 large eddy simulations have been carried out and 270 training samples are finally generated, which used around one million CPU

hours on high-performance computing (HPC) clusters. This high-fidelity database is used in this section to build an accurate and efficient surrogate model to predict the wind flow around wind turbine with the inflow wind profile  $U_{in}$  and the turbine yaw angle  $\gamma$  as the model input. The interested reader can refer to Ref. [28] for further details such as the mesh generation, the atmospheric boundary layer simulation, and the turbine specification.

Specifically, the model input  $\mu$  is specified as the combination of the wind speed at 32 uniformly distributed points along the inflow boundary and the value of turbine yaw angle. Thus the dimension of  $\mu$  is 33. The flow field  $U$  is specified as the combination of the streamwise velocity field  $U_x$  and spanwise velocity field  $U_y$  at the 32 × 32 uniform grid points in the flow domain shown in Fig. 2. Thus the dimension of  $U$  is 32 × 32 × 2. We mention that the considered fluid system is governed by the NS equations where the turbine rotors are modeled as actuator lines in the momentum equations. Thus the turbine parameters such as  $\gamma$  appear in the governing equations. Therefore, this case demonstrates that the proposed surrogate modeling method can accommodate flow parameters arising from different sources (here boundary conditions and governing equations) and can be used for the prediction of multi-channel flow fields.

### 3.2. Model training and validation

All the high-fidelity CFD data are first divided into 216 training samples and 54 test samples, then the surrogate model is trained based on the flow field data in the training samples. After training, its performance is evaluated by comparing the flow field predictions with the corresponding CFD data in the test samples which are assumed unavailable during the training process. The results for four randomly-selected test cases are shown in Fig. 4, including both the prediction by the developed CGAN model and the ground truth. The corresponding error distributions are shown in Fig. 5. The flow conditions for these four cases are:  $\gamma = -13.6^\circ$  and the upstream wake inflow (Case#1);  $\gamma = 16.5^\circ$  and the freestream inflow (Case#2);  $\gamma = -26.1^\circ$  and the freestream inflow (Case#3);  $\gamma = -1.1^\circ$  and the upstream wake inflow (Case#4). As can be seen, the flow field predictions, including both the streamwise velocity field  $U_x$  and the spanwise velocity field  $U_y$ , match with the corresponding LES results very well for all the cases. The main features of the flow field are captured accurately, such as the wake deflection with the turbine yaw angle, the wake recovery in the streamwise direction, the wake expansion in the spanwise direction, and the upstream wake's impact on the downstream flow field. The flow details are also predicted such as the fluctuations in the incoming wind, the flow acceleration on both sides of the turbines, the turbines' blockage effects on the upstream flow, and the yaw-induced spanwise velocity in the wake regions.

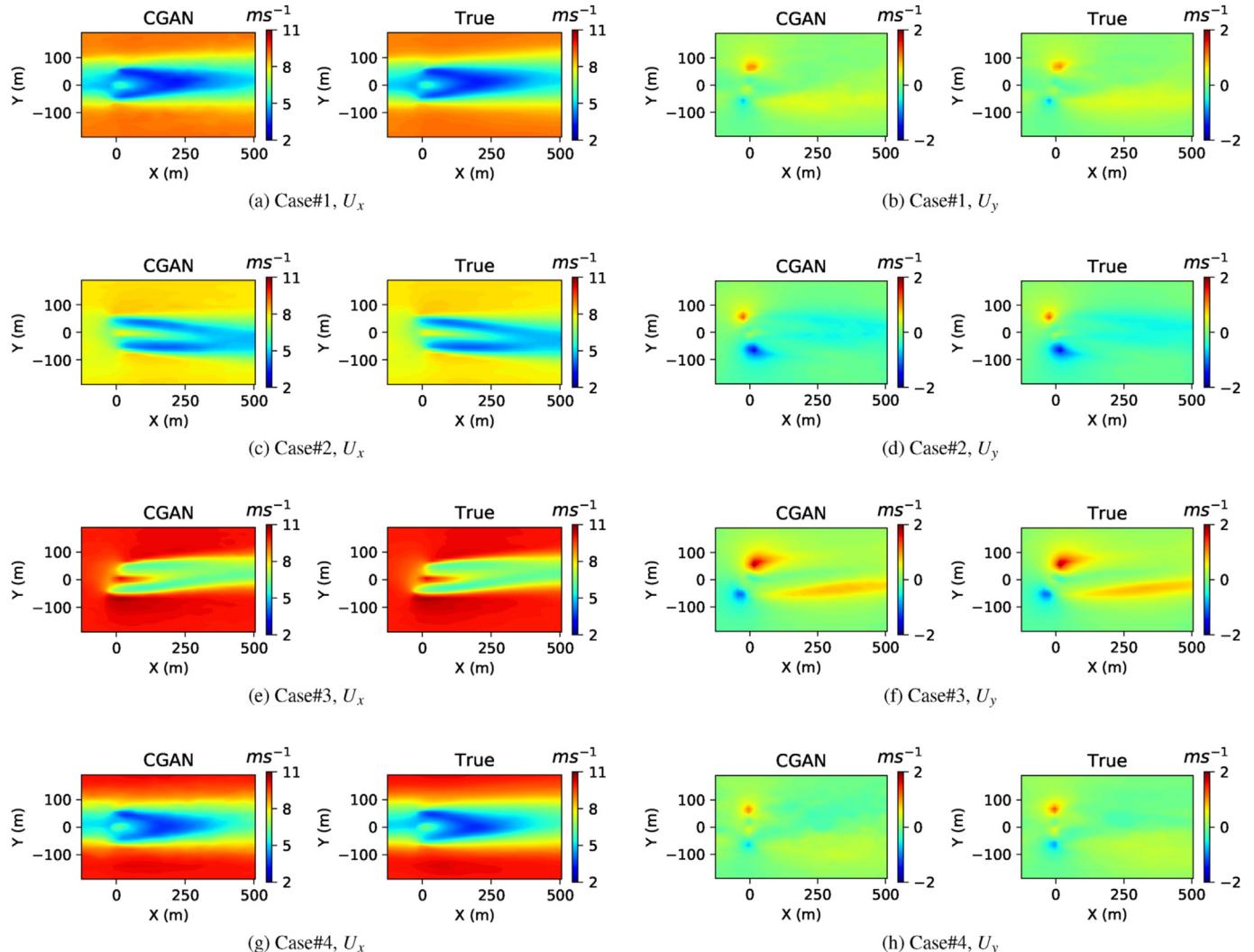
To further quantify the prediction accuracy, the root mean square errors (RMSE) of the CGAN model predictions for all the 54 test cases are calculated, which are defined as

$$\varepsilon_{U_x} = \sqrt{\frac{1}{N_{test} N_{grd}} \sum_{j=1}^{N_{test}} \sum_{i=1}^{N_{grd}} (U_{xi}^j - \hat{U}_{xi}^j)^2} \quad (6)$$

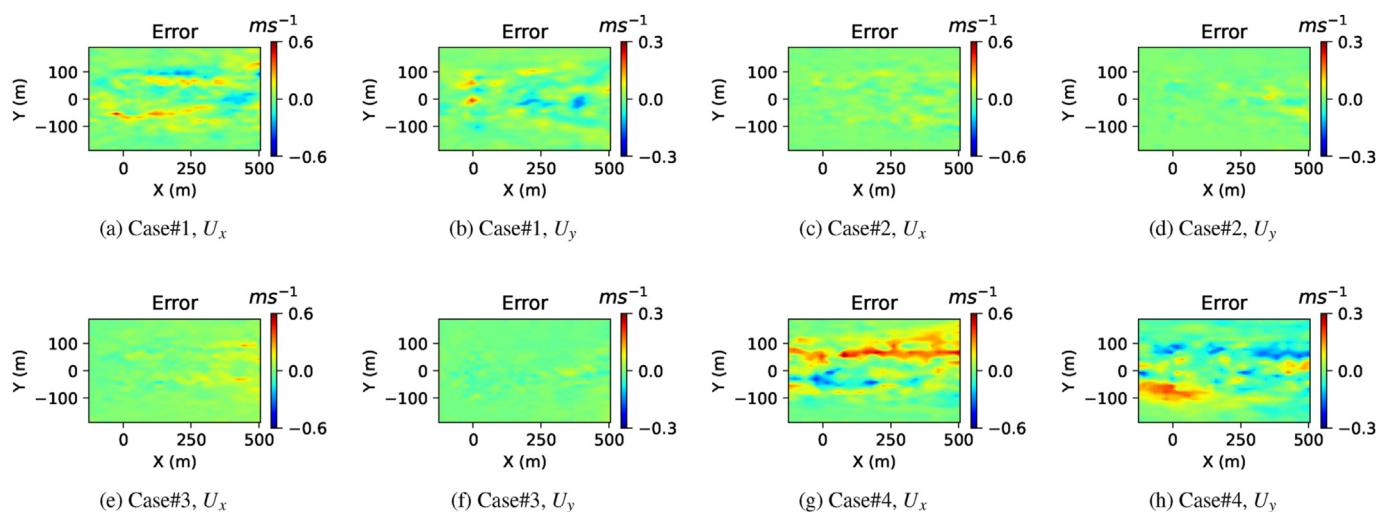
and

$$\varepsilon_{U_y} = \sqrt{\frac{1}{N_{test} N_{grd}} \sum_{j=1}^{N_{test}} \sum_{i=1}^{N_{grd}} (U_{yi}^j - \hat{U}_{yi}^j)^2} \quad (7)$$

for the streamwise and spanwise velocity field respectively. Here



**Fig. 4.** The comparisons between the predictions by the developed CGAN model and the corresponding true values for four randomly-selected test cases.



**Fig. 5.** The difference between the flow field predicted by the developed CGAN model and the corresponding true values for four randomly-selected test cases.

$N_{test}$  is the total number of test cases,  $N_{grd}$  is the dimension of the grid,  $U_{xi}^j$  and  $\hat{U}_{xi}^j$  represent the true and predicted values of the streamwise velocity at the  $i$ th grid point for the  $j$ th test case, and  $U_{yi}^j$  and  $\hat{U}_{yi}^j$  represent the true and predicted values of the spanwise velocity at the  $i$ th grid point for the  $j$ th test case. The results are given in Table 2. As shown, the RMSEs of the streamwise and spanwise velocity field predictions are  $0.102 \text{ ms}^{-1}$  and  $0.045 \text{ ms}^{-1}$  respectively, which are just 1.1 % and 1.0 % of the corresponding value ranges. We mention that the wake prediction with such accuracy is achieved with an online prediction time of just 0.002s, which is negligible compared to large-scale numerical simulations which will require several thousand CPU hours.

In addition, as the turbine wake evolves greatly in the downstream direction, the performance of the developed wake model is further examined by extracting the wind velocity profiles at various streamwise locations ranging from upstream of the turbine, near wake, to far wake. The results are given in Fig. 6 including both the prediction results and the corresponding ground truth. At the cases with freestream inflow, as shown in Fig. 6(c,e), the ‘flat’ inflow profile is observed upstream of the wind turbine ( $X = -1D$ ), then the ‘double-peak’ profile is predicted at the turbine location ( $X = 0D$ ) and in the near wake region ( $X = 1D$  and  $X = 2D$ ). The wake then recovers to the ‘bell-shape’ profile traveling further downstream. As for the cases with upstream wake inflow, as shown in Fig. 6(a,g), the ‘bell-shape’ profile is observed at upstream location ( $X = -1D$ ), then the wake development is observed similarly as the freestream inflow cases but with a clearly faster recovery to the ‘bell-shape’ profiles. The spanwise velocity profiles are shown in Fig. 6(b,d,f,h). As shown, the inducing of the spanwise velocity at turbine location ( $X = 0D$ ) is observed for all the cases. The yaw-induced spanwise velocity is clearly observed for the case where the turbine operates in strongly-yawed condition, as shown by Fig. 6(f). To further quantify the accuracy, the prediction RMSEs for the velocity profiles at various locations are calculated. For the streamwise velocity, the RMSEs are  $[0.064, 0.090, 0.098, 0.117, 0.109, 0.139] \text{ ms}^{-1}$  for  $X = [-1D, 0D, 1D, 2D, 3D, 4D]$  respectively. For the spanwise velocity, the RMSEs are  $[0.030, 0.050, 0.038, 0.045, 0.050, 0.063] \text{ ms}^{-1}$  for  $X = [-1D, 0D, 1D, 2D, 3D, 4D]$  respectively. It is concluded that the developed CGAN model is able to predict the wind turbine wake flows very accurately, at all the streamwise locations including both near wake and far wake, for both streamwise and spanwise velocity components, and at the cases with different inflow wind profiles.

### 3.3. Performance comparison with the POD-NN method

This subsection further demonstrates the performance of the developed CGAN wake model, by comparing it with a recently proposed surrogate modeling method called POD-NN [28,46,47]. In general, for the surrogate modeling of fluid systems, the high-dimensional flow field is usually reduced to its low-dimensional representations by a dimensionality reduction technique, then a supervised ML model is constructed to predict the reduced

coefficients with the flow parameters as the model input. Such approaches have been employed in various fluid applications recently [48,49]. In this subsection, the surrogate modeling of wind turbine wake flow is carried out by the POD-NN method, where the flow field is first reduced by the POD and then the POD coefficients are predicted by NN with the turbine yaw angle and the inflow velocity profile as the input. The hyper-parameters involved in the POD-NN method are tuned using the four-fold cross-validation technique. Specifically, for the results presented in this subsection, the number of the POD basis is set as 25, the L2 regularization is applied to the NN training with a regularization coefficient of 0.1, and the neuron number of the hidden layer is set as 50. We mention that for comparison purpose, the POD-NN model is trained and evaluated with the same training and test datasets as the CGAN model developed in previous sections. The prediction RMSEs by the POD-NN method are calculated and given in Table 2. As shown, the CGAN model is clearly more accurate in predicting the streamwise velocity field, while the prediction accuracy of both models is similar in predicting the spanwise velocity, as the spanwise velocity component is much simpler than the streamwise component. In addition, for the POD-NN, the dimensionality reduction errors for the spanwise and streamwise velocity are  $0.021 \text{ ms}^{-1}$  and  $0.039 \text{ ms}^{-1}$  respectively, while there is no dimensionality reduction error for the CGAN method. It is also worth mentioning that careful tuning of the hyper-parameters (in particular the number of the POD basis) is needed to achieve such accuracy for the POD-NN, while it is not the case for the CGAN model where dimensionality reduction is not needed. The results in this subsection thus demonstrate the accuracy and robustness of the developed CGAN wake model compared with the POD-NN.

### 3.4. Generalization performance

To systematically evaluate the generalization performance of the developed CGAN model, in this subsection, a series of flow field predictions (including the flow scenarios that are distinct from the training dataset) are carried out. In particular, the parametric study considers a series of turbine yaw angles, freestream wind speeds, and upstream wake magnitudes.

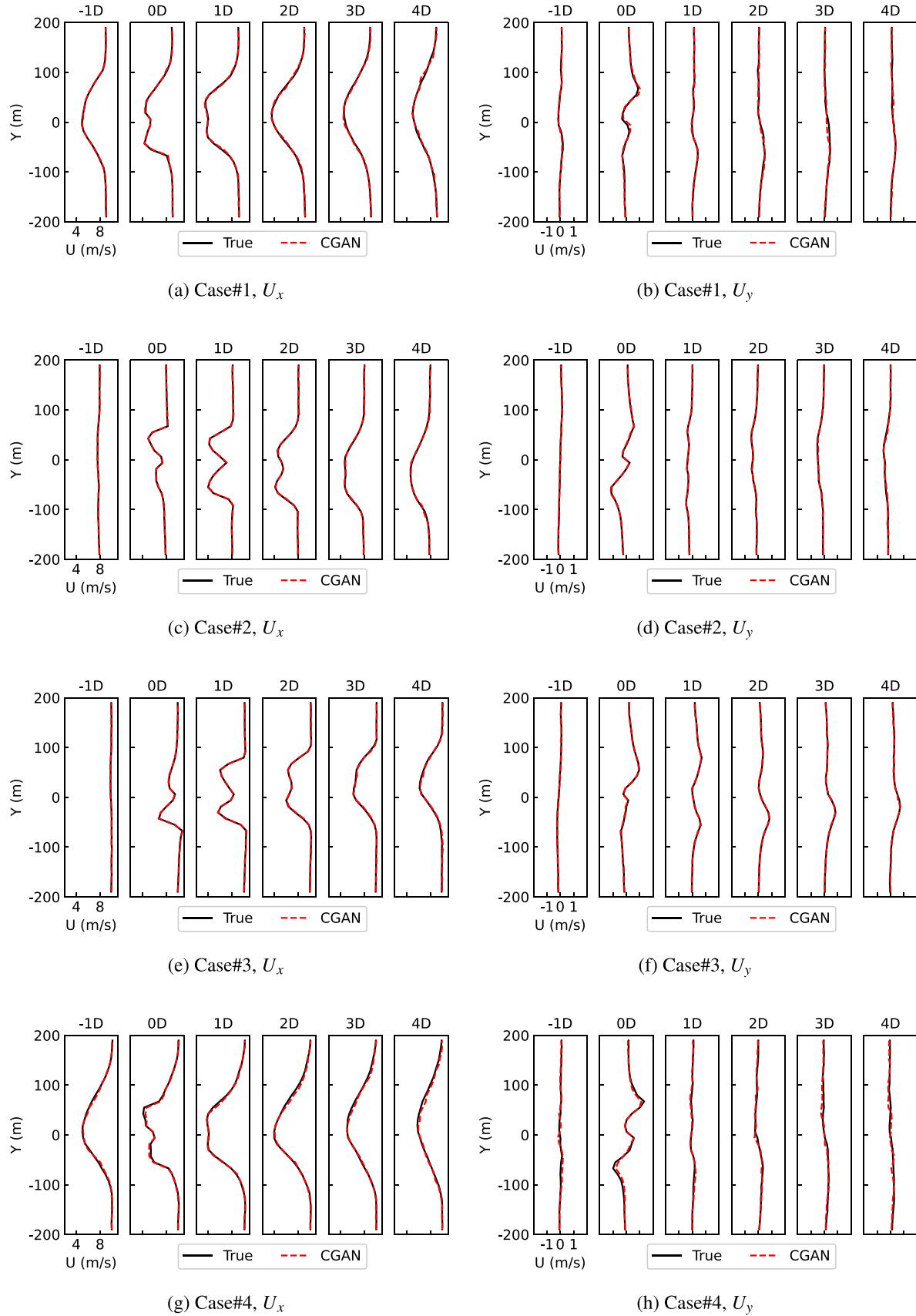
First, a series of turbine yaw angles and freestream wind speeds are considered. For this set of predictions, the inflow wind profile  $U_{in}$  is specified as constant values (i.e. the freestream wind speed) along the inflow boundary. The prediction results are given in Fig. 7. Each column in Fig. 7 shows the flow field predicted with the same inflow profiles and different yaw angles i.e.  $[-30^\circ, -20^\circ, -10^\circ, 0^\circ, 10^\circ, 20^\circ, 30^\circ]$ , while each row shows the flow field predicted with the same yaw angle and different freestream wind speeds i.e.  $[7, 8, 9, 10] \text{ ms}^{-1}$ . As can be seen, the wake deflection is captured successfully with the change of the turbine yaw angle, and the velocity magnitude in the whole domain is predicted well with the change of the freestream wind speed. We emphasize that the training dataset, as described in Section 3.1, includes only the cases with the freestream wind speed of  $8 \text{ ms}^{-1}$ ,  $9 \text{ ms}^{-1}$  and  $10 \text{ ms}^{-1}$ . Thus the prediction of the  $7 \text{ ms}^{-1}$  cases, i.e. the first column of Fig. 7, demonstrates that the CGAN wake model captures the qualitative flow features (such as yaw effects) successfully even for unknown flow scenarios.

Next, a series of upstream wake inflows are considered. Specifically, the Gaussian profile, one of the most popular wake profiles in analytical wake modeling, is used to specify the inflow wind profile  $U_{in}$  as

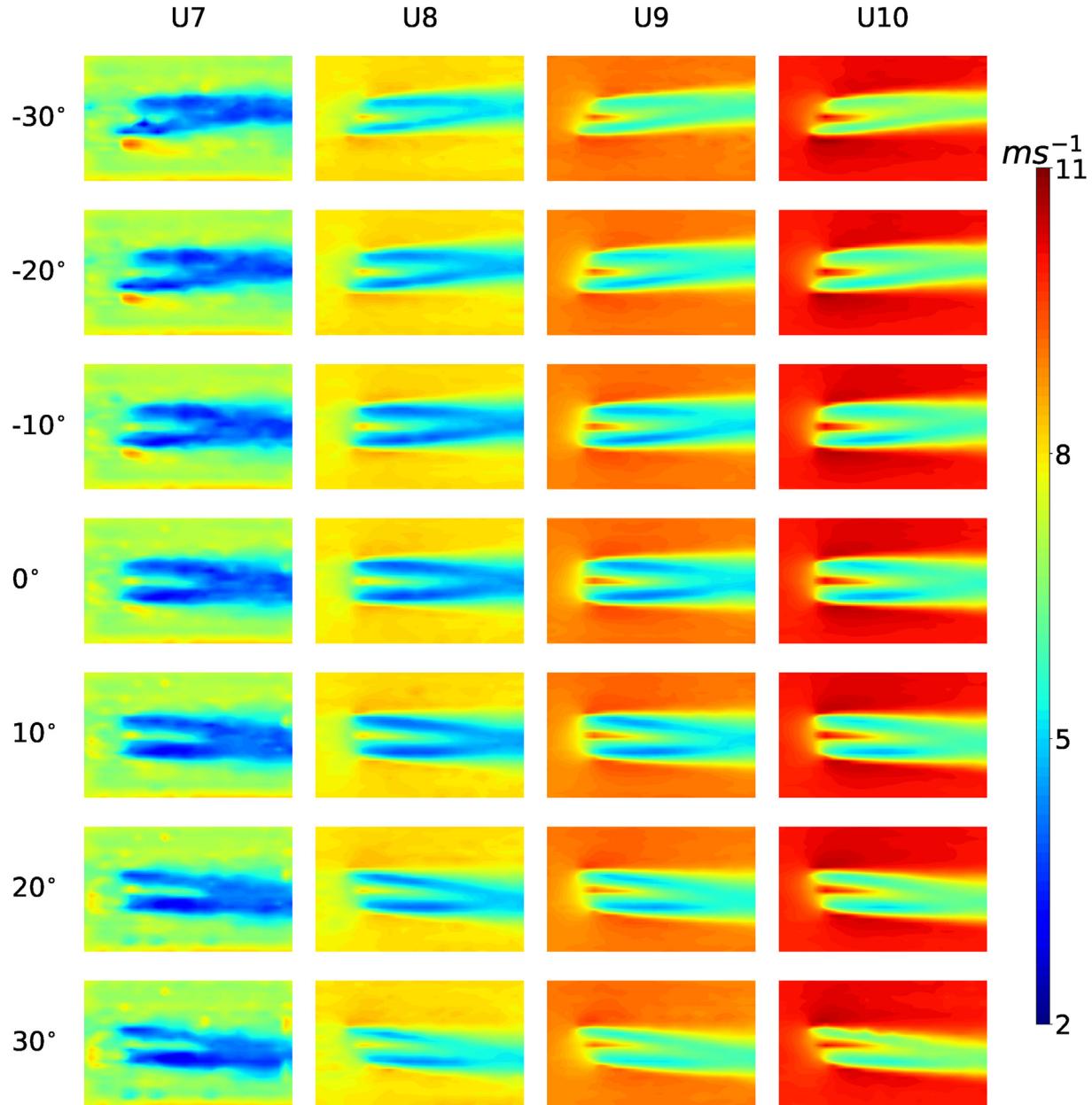
**Table 2**

The RMSEs of the flow field predictions by the developed CGAN model. The results by the POD-NN method are also included for comparisons.

Model	Quantity	Value range	RMSE (% of range)
CGAN	$U_x (\text{ms}^{-1})$	$[1.92, 10.84]$	0.102 (1.1 %)
	$U_y (\text{ms}^{-1})$	$[-2.25, 2.14]$	0.045 (1.0 %)
POD-NN	$U_x (\text{ms}^{-1})$	$[1.92, 10.84]$	0.131 (1.5 %)
	$U_y (\text{ms}^{-1})$	$[-2.25, 2.14]$	0.045 (1.0 %)



**Fig. 6.** The velocity profiles predicted by the developed CGAN model at various streamwise locations for four randomly-selected test cases. The corresponding true values are also shown for comparisons.

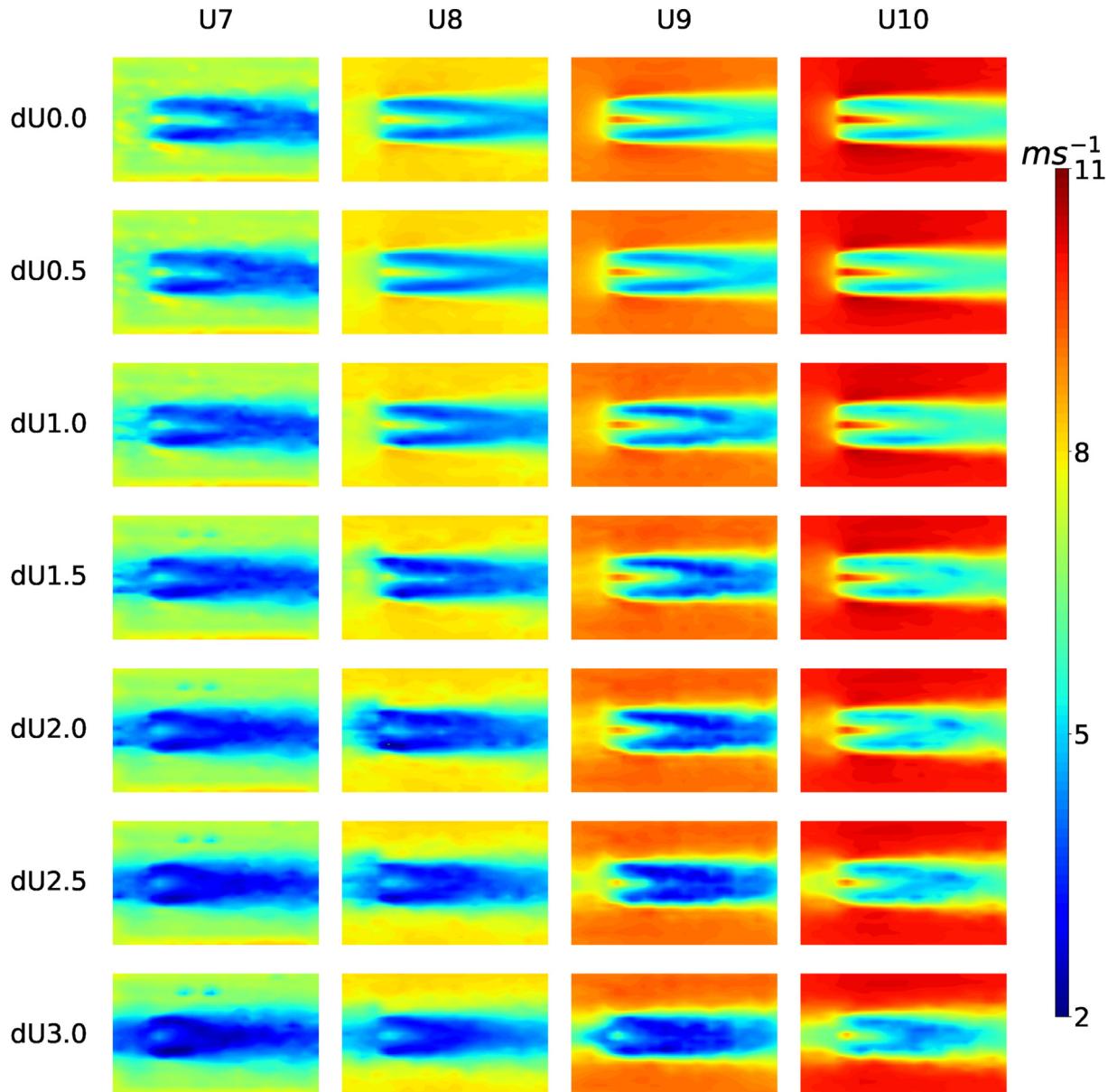


**Fig. 7.** The flow fields predicted by the developed CGAN model at a series of turbine yaw angles and freestream wind speeds.

$$U_{in} = U_0 - dU \exp\left(-\frac{y^2}{2\sigma_g^2}\right), \quad (8)$$

where  $U_0$  represents the freestream wind speed,  $dU$  represents the magnitude of the upstream wake,  $y$  is the spanwise coordinate, and  $\sigma_g$  characterizes the wake width. The value of  $\sigma_g$  is set as 50 m in the following predictions, which leads to a reasonable wake width. The

prediction results are given in Fig. 8, where each column shows the flow field predicted with the same freestream wind speed  $U_0$  and different upstream wake magnitudes  $dU$  i.e. [0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0]  $ms^{-1}$ , while each row shows the flow field predicted with the same upstream wake magnitude  $dU$  and different freestream wind speed  $U_0$  i.e. [7, 8, 9, 10]  $ms^{-1}$ . As can be seen, the main features of the flow field are predicted well, where the far field is mainly influenced by the freestream wind speed and the wake

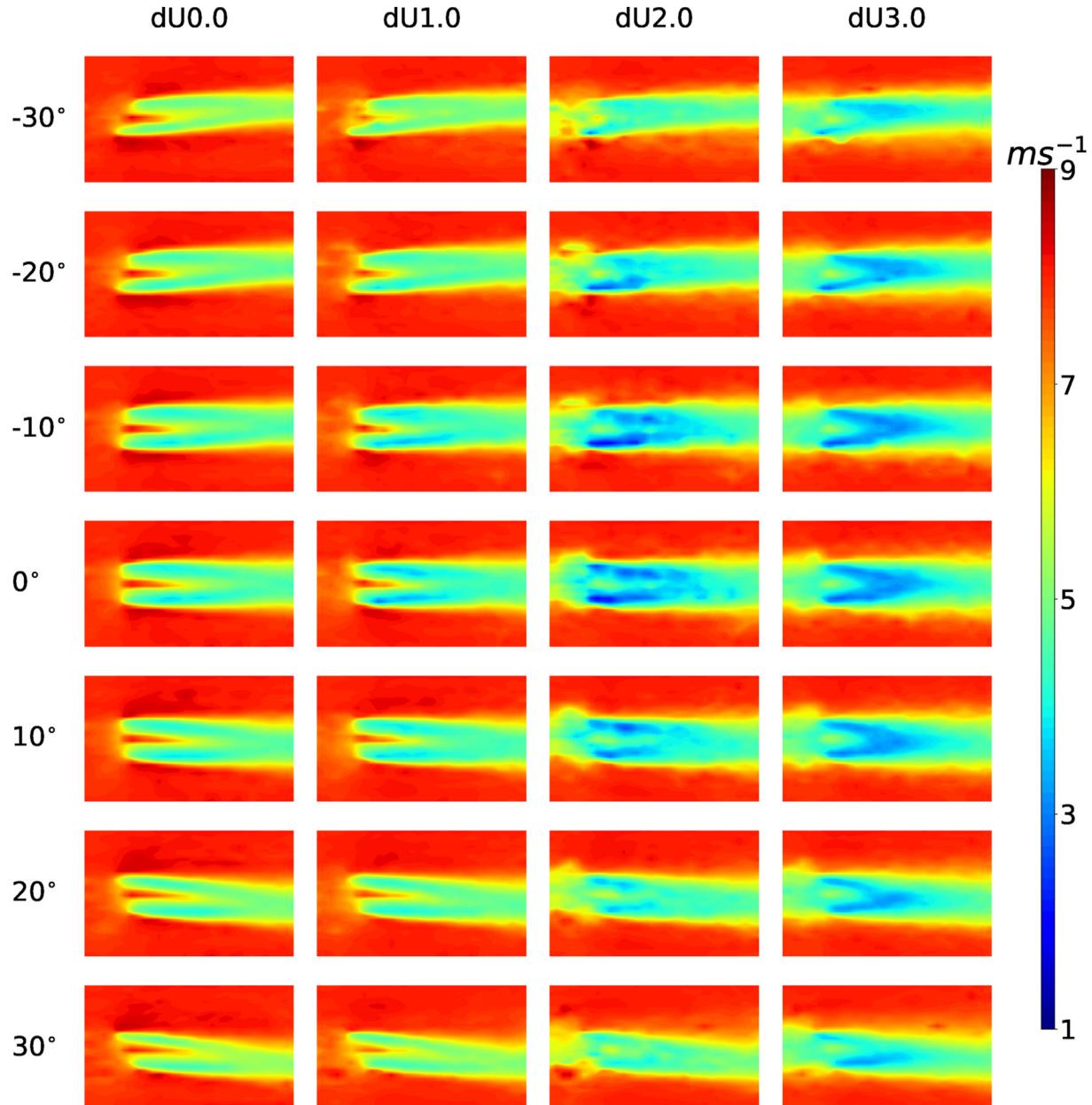


**Fig. 8.** The flow fields predicted by the developed CGAN model for a series of upstream wake magnitudes and freestream wind speeds.

regions are clearly influenced by the upstream wake magnitudes. We mention that the training dataset only includes the cases where the upstream wake magnitude is around  $3\text{--}4\text{ ms}^{-1}$ , while the CGAN wake model is shown to be able to predict the flow field with various upstream wake magnitudes. In addition, the 'bell-shape' inflow profiles in the training dataset are not Gaussian but the results shown in Fig. 8 are predicted with Gaussian profiles as the model input. This set of predictions thus demonstrate that the

developed CGAN model generalizes well to the inflow wind profiles that are not present in the training dataset.

Last but not least, to complete this parametric study, the predictions for a series of turbine yaw angles and upstream wake magnitudes are carried out. The results are shown in Fig. 9, where the freestream wind speed is set as  $8\text{ ms}^{-1}$ . As can be seen, the wake deflection and the wake deficit are predicted well with respect to the turbine yaw angles and the upstream wake magnitudes.

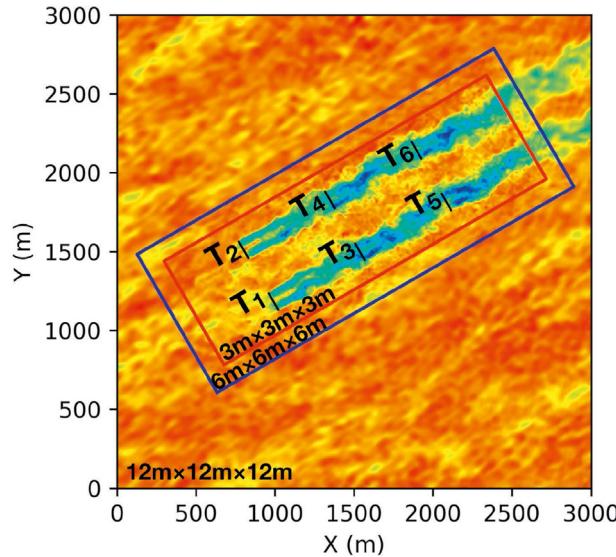


**Fig. 9.** The flow fields predicted by the developed CGAN model for a series of turbine yaw angles and upstream wake magnitudes.

In summary, the results in Figs. 7–9 fully demonstrate that the developed CGAN wake model generalizes well to unknown flow scenarios and learns the qualitative features of wind turbine wake flows successfully. It is concluded that the developed model can achieve efficient, accurate, and robust predictions of wind turbine wakes.

### 3.5. A case study

To further demonstrate the use of the developed CGAN wake model for wind farm applications, a case study for a wind farm is carried out in this subsection. The considered wind farm consists of six NREL 5 MW wind turbines and the turbine layout is illustrated in Fig. 10. For comparison purpose, the high-fidelity simulations



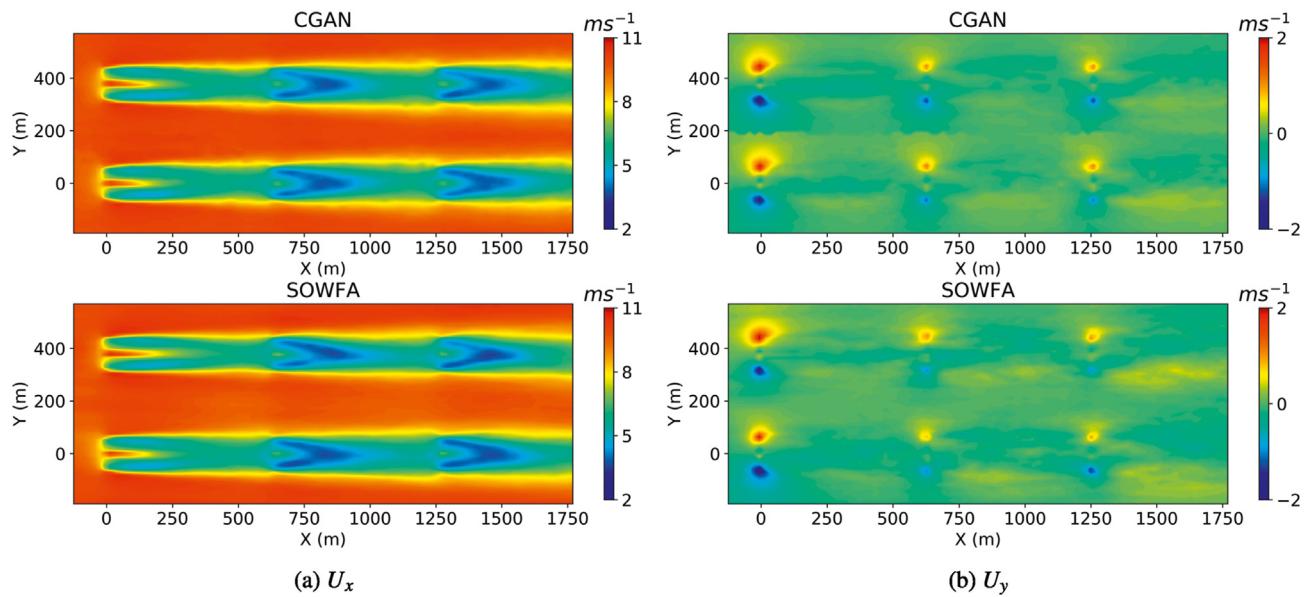
**Fig. 10.** The illustration of the considered wind farm layout and the mesh configuration for high-fidelity simulations.

using SOWFA are also carried out for this wind farm. The corresponding mesh configuration is shown in Fig. 10 where two-level mesh refinement is applied so that the mesh size in the turbine wake region is 3 m. The total number of mesh is around  $2.6 \times 10^7$ . The SOWFA simulations are carried out with a precursor atmospheric boundary layer simulation where the mean wind speed at hub height is set as  $10 \text{ ms}^{-1}$  and the turbulence intensity is set as 6 %. After wind farm simulations, the mean flow field at turbine hub height is extracted. In order to compare the CGAN model

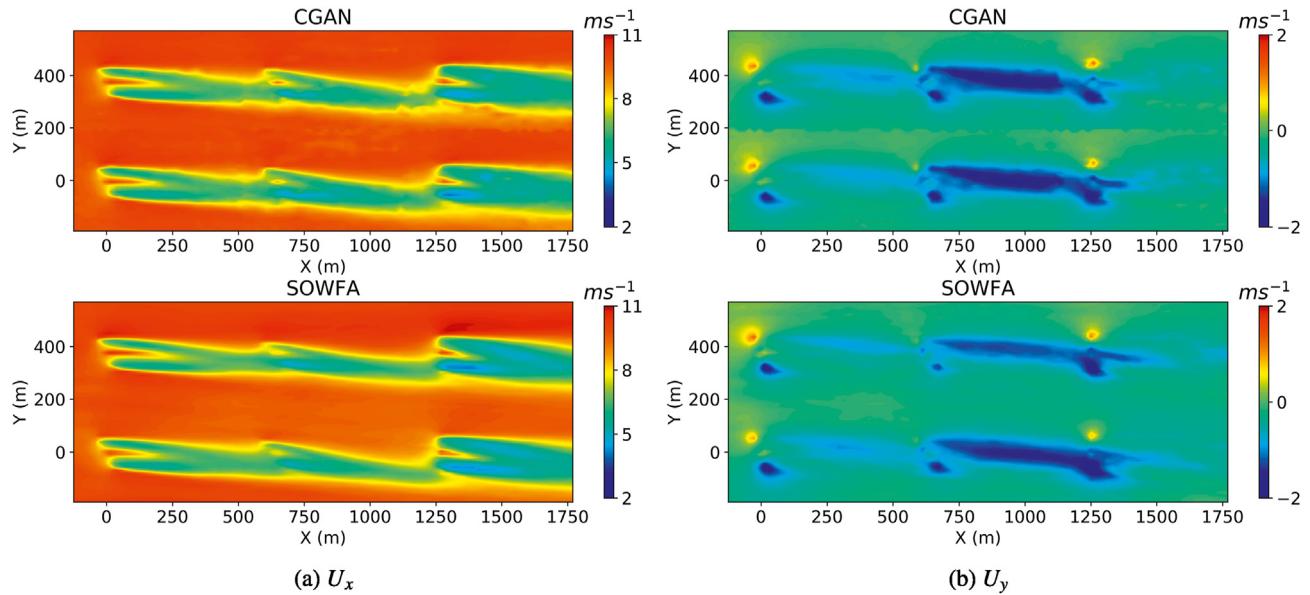
predictions with SOWFA results at the same wind conditions, the freestream wind profile is extracted from SOWFA and then used as the inflow wind conditions for the CGAN model.

The prediction of single turbine wakes has been demonstrated in previous subsections. The extension to wind farm wake prediction in this subsection follows the method proposed in Ref. [28]. Specifically, it is carried out by predicting single turbine wakes from upstream to downstream locations, where the inflow wind profiles for the downstream turbines are specified by the prediction results of the upstream turbine wakes. The flow field of the whole wind farm is finally obtained by combining the prediction results of all the single turbine wakes. The interested reader may refer to Ref. [28] for further details.

Two yaw control strategies are considered in this case study, i.e. the greedy case and the wake-steering case. The wind velocity fields, including both the streamwise and spanwise velocity, are predicted using both SOWFA and the developed CGAN model. For the greedy case, the yaw angles of all the wind turbines are set as  $0^\circ$ . The results are given in Fig. 11. As shown, the streamwise and spanwise velocity fields predicted by the CGAN model match well with SOWFA results. It is observed that the rear turbines operate in the full wakes generated by the upstream turbines. Then the wake-steering case is investigated, where the yaw angles of all the wind turbines are specified according to the optimized yaw angles reported in Ref. [17]. The yaw angles for T1, T2, T3, T4, T5 and T6 are  $[25.85, 25.15, 39.80, 39.75, 0.35, 0.45]$  degrees respectively. The results are given in Fig. 12. A great match is observed between the SOWFA and CGAN predictions, such as the yaw-induced wake deflections (see Fig. 12(a)) and the yaw-induced spanwise wind speed magnitudes (see Fig. 12(b)). As expected, the wakes generated by the front turbines are steered away from the rear turbines (i.e. T5 and T6) in this wake-steering case. We mention that the data used for training the CGAN model only includes the yaw angles in the range of  $[-30^\circ, 30^\circ]$ , while the yaw angles of T3 and T4 in this case largely exceed this range. Therefore, the wake-steering case further



**Fig. 11.** The prediction results by the developed CGAN model and SOWFA for the greedy case.



**Fig. 12.** The prediction results by the developed CGAN model and SOWFA for the wake-steering case.

demonstrates the robustness of the developed CGAN wake model in the scenarios of extrapolated turbine yaw angles.

#### 4. Conclusions

In this work, a surrogate modeling method for general fluid flows was proposed based on the deep convolutional conditional generative adversarial network (DC-CGAN). The proposed method was then applied to develop a novel wind farm wake model for the wake predictions under various inflow conditions and turbine yaw settings. The performance of the developed wake model was extensively evaluated, as summarized below.

- (1) The validation study against high-fidelity CFD results showed that the developed model was able to predict the wind turbine wake flows very accurately and efficiently, at all the streamwise locations including both near wake and far wake, for both streamwise and spanwise velocity components, and at the cases with different inflow wind profiles. In particular, the prediction errors for the streamwise and spanwise velocity fields were just  $0.102 \text{ ms}^{-1}$  and  $0.045 \text{ ms}^{-1}$  respectively, while the online computation time was only 0.002s. The main wake features, including the wake deflection with the turbine yaw angle, the wake recovery in the streamwise direction, the wake expansion in the spanwise direction, the flow acceleration on both sides of the turbines, and the yaw-induced spanwise velocity in the wake regions, were all predicted very accurately.
- (2) The parametric study for a set of turbine yaw angles, free-stream wind speeds, and upstream wake magnitudes was carried out and the results showed that the developed CGAN model generalized well to the flow scenarios that were not present in the training dataset, and that the developed model successfully learned the qualitative features of wind turbine wake flows.
- (3) The case study for a wind farm was carried out using both the developed model and the high-fidelity LES model. The comparison study showed a good agreement between the developed model and the LES model, including the predictions for both the streamwise and the spanwise velocity

fields. This case study fully demonstrated the performance of the developed model in wind farm applications.

Future works may involve the surrogate modeling of three-dimensional/dynamic wind farm wake flows. Another direction is to explore the use of the developed model in the optimal control design of wind farms.

#### Credit author statement

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

#### 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.

#### Acknowledgments

This work has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie grant agreement No 765579. It was also supported by the UK Engineering and Physical Sciences Research Council (grant number: EP/S000747/1) and the Scientific Computing Research Technology Platform (SCRTP) at the University of Warwick.

#### References

- [1] Adaramola M, Krogstad P-Å. Experimental investigation of wake effects on wind turbine performance. *Renew Energy* 2011;36(8):2078–86.
- [2] Boersma S, Doekemeijer B, Gebräad PM, Fleming PA, Annoni J, Scholbrock AK, et al. A tutorial on control-oriented modeling and control of wind farms. In: 2017 American control conference (ACC). IEEE; 2017. p. 1–18.
- [3] Jensen NO. A note on wind generator interaction. 1983.
- [4] Katic I, Højstrup J, Jensen NO. A simple model for cluster efficiency. In:

- Ragazzi A, editor. European wind energy association conference and exhibition; 1987.
- [5] Frandsen S, Barthelmie R, Pryor S, Rathmann O, Larsen S, Højstrup J, et al. Analytical modelling of wind speed deficit in large offshore wind farms. *Wind Energy* 2006;9(1–2):39–53.
- [6] G. C. Larsen, H. M. Aagaard, F. Bingöl, J. Mann, S. Ott, J. N. Sørensen, et al., Dynamic wake meandering modeling. 2007.
- [7] Boersma S, Doekemeijer B, Vali M, Meyers J, Wingerden J-W v. A control-oriented dynamic wind farm model: Wfsm. *Wind Energy Sci* 2018;3(1): 75–95.
- [8] Lopez D, Kuo J, Li N. A novel wake model for yawed wind turbines. *Energy* 2019;178:158–67.
- [9] Witha B, Steinfeld G, Heinemann D. High-resolution offshore wake simulations with the les model palm. In: Wind energy-impact of turbulence. Springer; 2014. p. 175–81.
- [10] Martínez-Tossas LA, Churchfield MJ, Leonardi S. Large eddy simulations of the flow past wind turbines: actuator line and disk modeling. *Wind Energy* 2015;18(6):1047–60.
- [11] Wu Y-T, Porté-Agel F. Modeling turbine wakes and power losses within a wind farm using les: an application to the horns rev offshore wind farm. *Renew Energy* 2015;75:945–55.
- [12] Calaf M, Meneveau C, Meyers J. Large eddy simulation study of fully developed wind-turbine array boundary layers. *Phys Fluid* 2010;22(1):015110.
- [13] Meyers J, Meneveau C. 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.
- [14] Nilsson K, Ivanell S, Hansen KS, Mikkelsen R, Sørensen JN, Breton S-P, et al. Large-eddy simulations of the lillgrund wind farm. *Wind Energy* 2015;18(3): 449–67.
- [15] Lu H, Porté-Agel F. Large-eddy simulation of a very large wind farm in a stable atmospheric boundary layer. *Phys Fluids* 2011;23(6):065101.
- [16] Churchfield MJ, Lee S, Michalakes J, Moriarty PJ. A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics. *J Turbul* 2012;13:N14.
- [17] Gebraad P, Teeuwisse F, Van Wingerden J, Fleming PA, Ruben S, Marden J, et al. Wind plant power optimization through yaw control using a parametric model for wake effects—a cfd simulation study. *Wind Energy* 2016;19(1): 95–114.
- [18] Zhang J, Zhao X. A novel dynamic wind farm wake model based on deep learning. *Appl Energy* 2020;277:115552.
- [19] Zhang J, Zhao X. Quantification of parameter uncertainty in wind farm wake modeling. *Energy* 2020;117065.
- [20] Bastankhah M, Porté-Agel F. A new analytical model for wind-turbine wakes. *Renew Energy* 2014;70:116–23.
- [21] Tian L, Zhu W, Shen W, Zhao N, Shen Z. Development and validation of a new two-dimensional wake model for wind turbine wakes. *J Wind Eng Ind Aerod* 2015;137:90–9.
- [22] Niayifar A, Porté-Agel F. Analytical modeling of wind farms: a new approach for power prediction. *Energies* 2016;9(9):741.
- [23] Dou B, Guala M, Lei L, Zeng P. Wake model for horizontal-axis wind and hydrokinetic turbines in yawed conditions. *Appl Energy* 2019;242:1383–95.
- [24] Ge M, Wu Y, Liu Y, Li Q. A two-dimensional model based on the expansion of physical wake boundary for wind-turbine wakes. *Appl Energy* 2019;233: 975–84.
- [25] Sun H, Yang H. Study on an innovative three-dimensional wind turbine wake model. *Appl Energy* 2018;226:483–93.
- [26] Sun H, Gao X, Yang H. Validations of three-dimensional wake models with the wind field measurements in complex terrain. *Energy* 2019;189:116213.
- [27] Kabir IFSA, Safiyullah F, Ng E, Tam VW. New analytical wake models based on artificial intelligence and rivaling the benchmark full-rotor cfd predictions under both uniform and abl inflows. *Energy* 2019;116761.
- [28] Zhang J, Zhao X. Machine-learning-based surrogate modeling of aerodynamic flow around distributed structures. *AIAA J* 2021;59(3):868–79.
- [29] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence N, Weinberger KQ, editors. Advances in neural information processing systems, vol. 27. Curran Associates, Inc.; 2014. p. 2672–80.
- [30] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784.
- [31] A. Radford, L. Metz, S. Chintala, 2015. Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434.
- [32] Isola P, Zhu J-Y, Zhou T, Efros AA. Image-to-image translation with conditional adversarial networks. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE; 2017.
- [33] M. Arjovsky, S. Chintala, L. Bottou, 2017. Wasserstein gan, arXiv preprint arXiv:1701.07875.
- [34] Zhu J-Y, Park T, Isola P, Efros AA. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: 2017 IEEE international conference on computer vision (ICCV). IEEE; 2017.
- [35] Xie Y, Franz E, Chu M, Thuerey N. Tempogan: a temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Trans Graph* 2018;37(4): 1–15.
- [36] Deng Z, He C, Liu Y, Kim KC. Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework. *Phys Fluids* 2019;31(12):125111.
- [37] Cheng M, Fang F, Pain C, Navon I. Data-driven modelling of nonlinear spatio-temporal fluid flows using a deep convolutional generative adversarial network. *Comput Methods Appl Mech Eng* 2020;365:113000.
- [38] A. Subramanian, M. L. Wong, R. D. Borker, S. Nimmagadda, S. K. Lele, 2020. Turbulence enrichment using physics-informed generative adversarial networks, arXiv preprint arXiv:2003.01907.
- [39] S. Ioffe, C. Szegedy, 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167.
- [40] NWTC information portal (SOWFA). <https://nwtc.nrel.gov/SOWFA/>; 2014.
- [41] Churchfield M, Lee S, Moriarty P, Martinez L, Leonardi S, Vijayakumar G, et al. A large-eddy simulations of wind-plant aerodynamics. In: 50th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition. American Institute of Aeronautics and Astronautics; 2012.
- [42] Fleming P, Gebräad PM, Lee S, van Wingerden J-W, Johnson K, Churchfield M, et al. Simulation comparison of wake mitigation control strategies for a two-turbine case. *Wind Energy* 2014;18(12):2135–43.
- [43] D. P. Kingma, J. Ba, 2014. Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.
- [44] Chollet F, et al. Keras. <https://github.com/fchollet/keras/>; 2015.
- [45] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, et al., 2016. Tensorflow: large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467.
- [46] Hesthaven J, Ubbiali S. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *J Comput Phys* 2018;363:55–78.
- [47] Swischuk R, Mainini L, Peherstorfer B, Willcox K. Projection-based model reduction: formulations for physics-based machine learning. *Comput Fluid* 2019;179:704–17.
- [48] Wang Q, Hesthaven JS, Ray D. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. *J Comput Phys* 2019;384:289–307.
- [49] Guo M, Hesthaven JS. Data-driven reduced order modeling for time-dependent problems. *Comput Methods Appl Mech Eng* 2019;345:75–99.