

## Article

# Time-Averaged Wind Turbine Wake Flow Field Prediction Using Autoencoder Convolutional Neural Networks

Zexia Zhang <sup>1</sup>, Christian Santoni <sup>1</sup>, Thomas Herges <sup>2</sup>, Fotis Sotiropoulos <sup>3</sup> and Ali Khosronejad <sup>1,\*</sup><sup>1</sup> Department of Civil Engineering, Stony Brook University, Stony Brook, NY 11794, USA; zexia.zhang@stonybrook.edu (Z.Z.); christian.santoni@stonybrook.edu (C.S.)<sup>2</sup> Wind Energy Technologies, Sandia National Laboratories, Albuquerque, NM 87185, USA; therges@sandia.gov<sup>3</sup> Mechanical & Nuclear Engineering Department, Virginia Commonwealth University, Richmond, VA 23284, USA; sotiropoulosf@vcu.edu

\* Correspondence: ali.khosronejad@stonybrook.edu

**Abstract:** A convolutional neural network (CNN) autoencoder model has been developed to generate 3D realizations of time-averaged velocity in the wake of the wind turbines at the Sandia National Laboratories Scaled Wind Farm Technology (SWiFT) facility. Large-eddy simulations (LES) of the SWiFT site are conducted using an actuator surface model to simulate the turbine structures to produce training and validation datasets of the CNN. The simulations are validated using the SpinnerLidar measurements of turbine wakes at the SWiFT site and the instantaneous and time-averaged velocity fields from the training LES are used to train the CNN. The trained CNN is then applied to predict 3D realizations of time-averaged velocity in the wake of the SWiFT turbines under flow conditions different than those for which the CNN was trained. LES results for the validation cases are used to evaluate the performance of the CNN predictions. Comparing the validation LES results and CNN predictions, we show that the developed CNN autoencoder model holds great potential for predicting time-averaged flow fields and the power production of wind turbines while being several orders of magnitude computationally more efficient than LES.

**Keywords:** convolutional neural network; wind turbine; wake flow predictions; large-eddy simulation



**Citation:** Zhang, Z.; Santoni, C.; Herges, T.; Sotiropoulos, F.; Khosronejad, A. Time-Averaged Wind Turbine Wake Flow Field Prediction Using Autoencoder Convolutional Neural Networks. *Energies* **2022**, *15*, 41. <https://doi.org/10.3390/en15010041>

## Academic Editors:

Luis Hernández-Callejo,  
Sergio Nesmachnow and  
Sara Gallardo Saavedra

Received: 15 November 2021

Accepted: 17 December 2021

Published: 22 December 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In a wind farm, turbine wake interactions cause power losses and may increase fatigue loads on downwind wind turbines [1,2]. Therefore, the accurate prediction of turbine wakes is an important consideration in wind farm layout optimization, which can improve the efficiency of power production and reduce the overall levelized cost of energy. As a result, extensive efforts have been made on analytical and numerical models for the estimation of turbines wake [3–6].

Due to the simplicity and low computational cost, engineering models are widely used to predict wake flows and optimize wind farm power production, especially in industrial applications. The very first and extensively studied model was proposed by Jensen [7]. This model was derived from mass conservation, assuming a top-hat shape distribution of velocity deficit in the wake. However, the top-hat wake shape assumption is an oversimplification of the actual wake flow, which can be represented more accurately by a Gaussian distribution [8–10]. Furthermore, more complex real-life characteristics of wake flows have also been considered to improve the accuracy and flexibility of the Gaussian models, including the double-Gaussian type velocity profile of the near wake [11,12], three-dimensional effects [13,14], more accurate models for turbulence intensities [15], wind turbine yaw offset [16], atmospheric stability, and Coriolis force [17]. Although these models are efficient, the accuracy varies significantly from case to case [18,19], especially in the near wake region [12,20]. In addition, wake overlapping effects are not accurately described, as shown by Archer et al. [21].

Compared to engineering models, computational fluid dynamic (CFD) models can provide a better physics-based description of the dynamics of the turbine wakes, such as wake meandering [22–24] and the effects of atmospheric stability [25]. Moreover, some CFD models can even take into account the effect of complex terrain topology [26–28] in addition to the turbine tower and the nacelle [29,30]. The prediction of the velocity deficit and turbulent kinetic energy using the CFD methods is more accurate than engineering models [31]. However, the CFD methods employing LES are computationally expensive, and their use in wind farm optimization is becoming prohibitively expensive.

The development of machine learning and artificial intelligence has encouraged researchers to explore data-driven models to predict the wake and power production of turbines in a wind farm. For example, Japar et al. [32] used five different machine-learning methods, i.e., linear regression, linear regression with feature engineering, nonlinear regression, artificial neural network (ANN), and support vector regression (SVR), to estimate wind turbine power production based on free stream wind speed, wind direction and the turbine position in the wind farm. Although the more elaborate models, i.e., ANN and SVR, have higher accuracy, they slightly deviate from the measured power production in the high wind speed case. Sun [33] developed an ANN to predict power production of wind farms that considers wake effects for varying wind direction, wind speed, and yaw angle. The trained model successfully predicted the power production and was used to optimize the yaw angle of each wind turbine. However, these power production models need a large amount of input parameter combinations for the training and may only be applied to a particular wind farm. When considering the effects of more parameters, for example, turbulence kinetic energy, the whole neural network has to be retrained.

Other data-driven machine learning models have focused on the prediction of the velocity deficit of the wake. Wilson et al. [34] used Random Forest and Multilayer Perceptron (MLP) models to interpolate and predict wind velocity in the turbine wake, but it cannot be applied in different wind fields. Ti et al. [35,36] developed an ANN to predict the velocity deficit and turbulent kinetic energy field in the turbine wake from the incoming wind velocity and turbulence intensity. However, the method requires a large amount of CFD simulation results for training. Yang [37] developed a neural network model to predict the instantaneous position of the meandering turbine wake, using the upwind velocity, turbine torque, and turbine thrust as input features. Zhang and Zhao [38] proposed a neural network combining different dimensionality reduction techniques to predict the velocity field of distributed fluid systems and applied it successfully to predict the flow field of both a single turbine and turbine arrays. King et al. [39] proposed a Gaussian Process (GP) model to correct wind turbine flow field predictions from low-fidelity models, e.g., RANS model, to high-fidelity models, e.g., LES model. Ali et al. [40] used a Long-Short Term Memory (LSTM) model to successfully predict wind velocity fluctuation at specific locations in the turbine wake for a long time period. Renganathan et al. [41] combined an MLP and GP with a Convolutional Neural Network (CNN) decoder to map the wind turbine operation parameters, such as inflow wind speed, turbulent intensity, turbine power generation, atmospheric-boundary layer (ABL) Richardson number, rotor angular speed, and pitch angle, to the wake flow field. In addition to wake reconstruction, data-driven methods have also been used to identify and characterize turbine wakes. Aird et al. [42] developed a mask Region based Convolutional Neural Network (R-CNN) model that identifies turbine wakes in Lidar scan images with high accuracy, even with some missing data points, and is also able to character wake shapes in its forming and dissipating.

Despite these contributions, the accuracy of the existing algorithms for velocity field predictions varies with flow conditions and wind farm layouts, limiting their application for wind farm optimization. In this study, we develop a CNN autoencoder model for generating 3D realizations of time-averaged turbulent wake flow of wind turbines at the Sandia National Laboratories Scaled Wind Farm Technology (SWiFT) site in Lubbock, Texas. The site includes three Vestas V27 wind turbines to investigate the performance of the downwind turbine versus the upwind turbine wake. SpinnerLidar measurements of the

upwind turbine wake at the SWiFT facility have been used to validate the LES results of our in-house virtual flow simulator code, Virtual Flow Simulator (VFS-Wind model). After these validation comparisons, a series of LESs of the SWiFT site were conducted to produce instantaneous and time-averaged flow field results to train and test the CNN. Subsequently, the so-trained CNN was employed to predict the time-averaged flow field of new wind conditions. A data augmentation technique is employed to handle the location sensitivity problems of the CNN. The CNN predictions for the validation test cases were compared against the simulation results of the separately done LES validation case not used in the CNN training. In addition, the predicted time-averaged flow field of wind turbines was used to predict the time-averaged power production of the wind turbines.

This paper is organized as follows. In Section 2, the governing equations of the numerical model and the computational details of the LES of the SWiFT site are presented. In Section 3, the CNN autoencoder algorithm is described, followed by the results and discussion in Section 4. Final remarks can be found in Section 5.

## 2. Numerical Methods

### 2.1. Governing Equations

Simulations of the SWiFT site are performed by the LES module of the in-house incompressible Navier-Stokes solver code—VFS-Wind model. In the VFS-Wind code, the incompressible turbulent flow is described by the filtered Navier-Stokes and continuity equation written in curvilinear coordinates given as [43]:

$$J \frac{\partial U^j}{\partial \xi^j} = 0, \quad (1)$$

$$\frac{1}{J} \frac{\partial U^i}{\partial t} = \frac{\xi_l^i}{J} \left( -\frac{\partial}{\partial \xi^j} (U^j u_l) \right) + \frac{1}{\rho} \frac{\partial}{\partial \xi^j} \left( \mu \frac{g^{jk}}{J} \frac{\partial u_l}{\partial \xi^k} \right) - \frac{1}{\rho} \frac{\partial}{\partial \xi^j} \left( \frac{\xi_l^i p}{J} \right) - \frac{1}{\rho} \frac{\partial \tau_{lj}}{\partial \xi^j} + f_l, \quad (2)$$

where  $J = |\partial(\xi^1, \xi^2, \xi^3) / \partial(x_1, x_2, x_3)|$  is the determinant of the Jacobian of the geometric transformation of the Cartesian coordinates  $\{x_i\}$  and the generalized curvilinear coordinates  $\{\xi^i\}$ ,  $\xi_l^i = \partial \xi^i / \partial x_l$  are the transformation metrics,  $g^{jk} = \xi_l^j \xi_l^k$  are the components of the contravariant metric tensor,  $u_i$  is the  $i$ th Cartesian velocity component,  $U^i = (\xi_m^i / J) u_m$  is the contravariant volume flux,  $p$  is the pressure,  $\rho$  is the fluid density,  $\mu$  is the dynamic viscosity of the fluid,  $\tau_{ij}$  is the sub-grid stress tensor for LES, which is modeled using the dynamic Smagorinsky sub-grid scale (SGS) model [44], and  $f_l$  is the body force.

### 2.2. Numerics

In the VFS-wind code, the governing equations are discretized on a hybrid staggered/non-staggered grid in space. The convective terms are discretized using second-order accurate central differencing. For the divergence, pressure gradient, and viscous-like terms, the discretization used is the second-order accurate, three-point central differencing method [45]. The time derivatives are discretized using a second-order backward differencing scheme [46]. The discrete flow equations are time-integrated using an efficient, second-order accurate fractional step methodology in conjunction with a Jacobian-free, Newton-Krylov solver for the momentum equations and a GMRES solver enhanced with the multigrid method as a preconditioner for the Poisson equation.

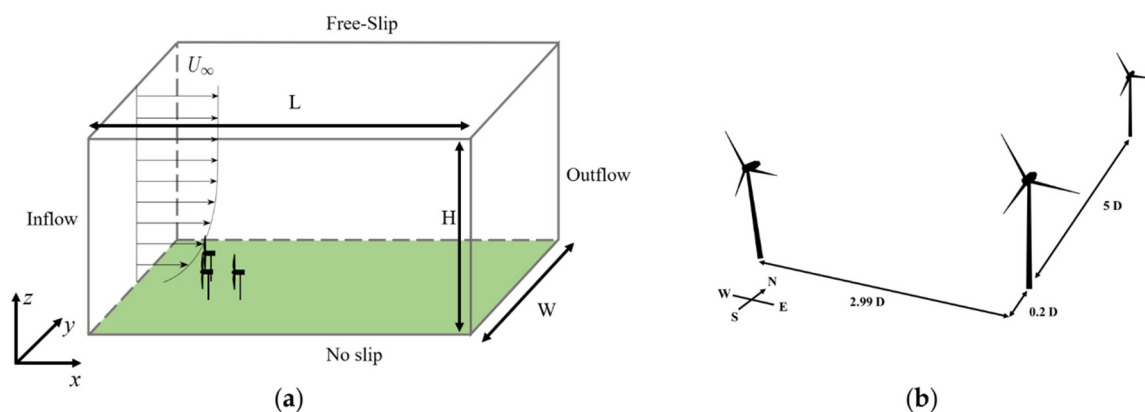
### 2.3. Actuator Surface Model

The wind turbine blades and nacelle are modeled using the actuator surface method developed by Yang and Sotiropoulos [47]. This method describes the flow field on the background Cartesian mesh and the wind turbine on the Lagrangian mesh following the actuator surfaces. Velocities on the actuator surfaces are interpolated from the background mesh using the smoothed discrete delta function proposed by Yang et al. [48]. Actuator surfaces of wind turbine blades are represented by chord lines along the radial direction.

Lift and drag forces on blades are calculated similarly to the actuator line model, using the blade element momentum theory. For the nacelle actuator surfaces, the normal component of force is calculated by reconstructing the wall-normal velocity near the actuator surface to satisfy the non-penetration constraint, and the tangential forces are computed as a function of the local incoming velocity and a friction coefficient that parameterizes the effects of near-wall turbulence and the effects of surface geometry. Then the counterforces to the flow field ( $f_i$  in Equation (2)) are calculated by distributing the forces on the blades to the background mesh using the above-mentioned interpolation method. Details of the actuator surface method can be found in Yang and Sotiropoulos [47].

### 3. Computational Details of SWiFT Site Simulation

The SWiFT facility, located in Lubbock, Texas, is an experimental site supported by the U.S Department of Energy to investigate turbine wakes and turbine-turbine interactions. It is comprised of three research-scale wind turbines and two meteorological towers (METs). The layout of wind turbines is as shown in Figure 1b. The wind turbines, Vestas V27s, have rotor diameters of  $D = 27$  m and hub heights of 32.1 m. A nacelle mounted Technical University of Denmark (DTU) SpinnerLidar is installed on turbine T1 to measure the turbine wake. The two METs are located upwind against the predominant wind direction to measure the atmospheric inflow. Details of the SWiFT facility can be found in Berg et al. [49].



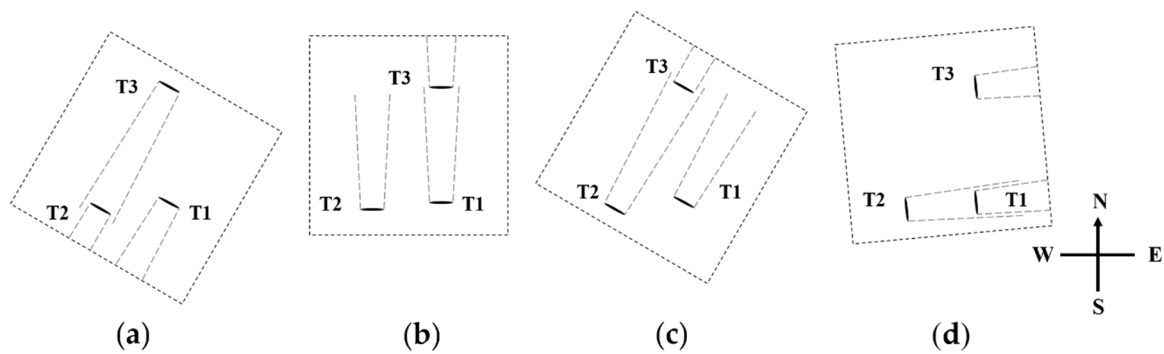
**Figure 1.** (a) is the diagram of the computational domain. (b) is the relative location of three turbines.

A series of LESs were performed to generate 3D flow fields of the SWiFT site for a variety of inflow conditions, which give rise to different turbine wake interaction configurations. The computational domain of the real-scale modeled SWiFT site is shown in Figure 1a. It has a length of  $L = 80D$ , a width of  $W = 24.9D$ , and a height of  $H = 37D$ . Free-slip boundary condition is applied to the top and periodic condition along the span-wise direction. A logarithmic law of wall boundary condition is applied to the ground, which is given by

$$u = u^* / \kappa \ln(z/z_0), \quad (3)$$

where  $u^*$  is the friction velocity,  $\kappa$  is the von Karman constant, and  $z_0 = 0.0037D$  is the surface roughness height. The outlet is given by the Neumann boundary condition, and the inlet is fed with a fully-developed turbulent flow generated by a precursor simulation as described below. Three wind turbines are located over  $8D$  downwind from the inlet. Each turbine has a hub height of  $h = 1.19D$  and a rotor diameter of  $D = 27$  m. Turbine 2 is  $2.99D$  west and  $0.2D$  south from turbine 1. Turbine 3 is  $5D$  north from turbine 1. The arrangement of the turbines is shown in Figure 1b. To generate different wake conditions, we conducted simulations for four wind directions ( $150^\circ$ ,  $0^\circ$ ,  $330^\circ$ , and  $274^\circ$ , taking south as  $0^\circ$ ) as shown in Figure 2. Although the wind directions do not perfectly align with the cardinal directions, for the sake of brevity, they would be referred to as North-East, South, South-West, and West, respectively, through the rest of the paper. The flow domain is rotated in different

wind directions to ensure the  $x$ -axis is always along the wind directions. In addition, five wind velocities are considered for each direction ( $U_\infty = 7, 9, 11, 13, 15$  m/s).



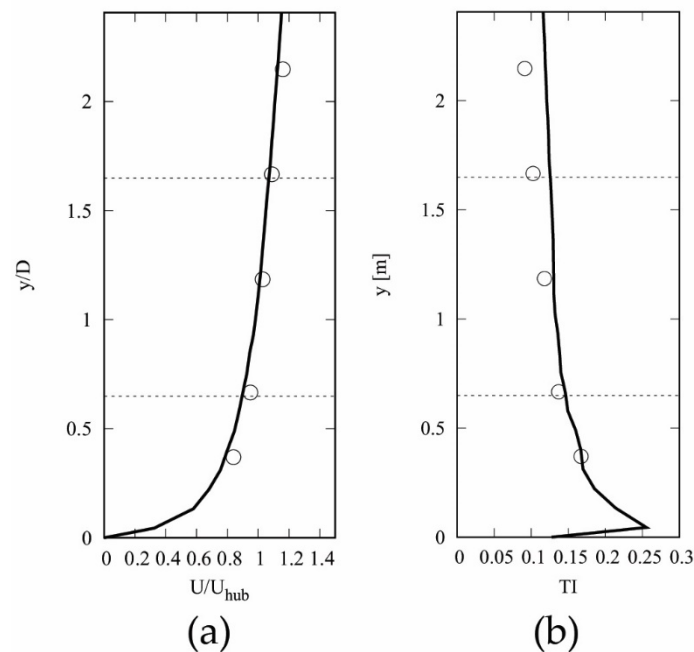
**Figure 2.** Diagram of wind turbine configurations in different wind directions; (a) north-east ( $150^\circ$ ), (b) south ( $0^\circ$ ), (c) south-west ( $330^\circ$ ), and (d) west ( $274^\circ$ ). Solid lines marked using T1, T2 and T3 represent the location of three turbines. Dashed lines represent the wake of each turbine.

The computational domain is discretized with a grid resolution of  $\Delta x = 0.177D$  and  $\Delta y = 0.089D$ , along the windwise and spanwise directions, respectively. A constant resolution of  $\Delta z = 0.089D$  was given along the wall-normal direction up to a height of  $7.46D$  and a grid stretching up to the top of the domain with a final resolution of  $\Delta z = 1.48D$ . Therefore, a uniform grid is obtained in the bottom area that the turbines are located. The details of the computational domain are shown in Table 1.

**Table 1.** Geometrical and wind characteristics of the SWiFT site model simulations used for the training and validation of the CNN.  $U_\infty$  is the free-flow velocity.  $Re$  is Reynolds number.  $D$  is the diameter of the turbine rotor ( $=27$  m).  $N_x$ ,  $N_y$ , and  $N_z$  are the number of computational grid nodes in windwise, spanwise, and vertical directions, respectively.  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$  are the special resolution in windwise, spanwise, and vertical directions, respectively.

$H$	$37D$	$N_x \times N_y \times N_z$	$451 \times 143 \times 281$
$W$	$24.9D$	$\Delta x$	$0.177D$
$L$	$80D$	$\Delta y$	$0.089D$
$U_\infty$ (m s $^{-1}$ )	7, 9, 11, 13, 15	$\Delta z$	$0.089\text{--}1.481D$
$Re$	$5.7 \times 10^8$	$\Delta t$ (s)	$2 \times 10^{-4} H/U_\infty$
$D$ (m)	27		

Precursor simulation of a neutral atmospheric boundary layer has been performed to prescribe the velocity at the inlet of the SWiFT site simulations. The precursor simulation has the same grid size and boundary conditions as the SWiFT site numerical domain. The initial transient of the simulation was discarded. After the mean kinetic energy of the computational domain reached steady state, velocities at a plane located in the center of the channel were saved at a  $\Delta t = 2.0 \times 10^{-4} H/U$ . The time-averaged velocity profile and turbulent intensity of the precursor simulation are shown in Figure 3.



**Figure 3.** Ten-minute time-averaged (a) velocity profile and (b) turbulence intensity; (○) met-tower measurements and (—) precursor simulation.

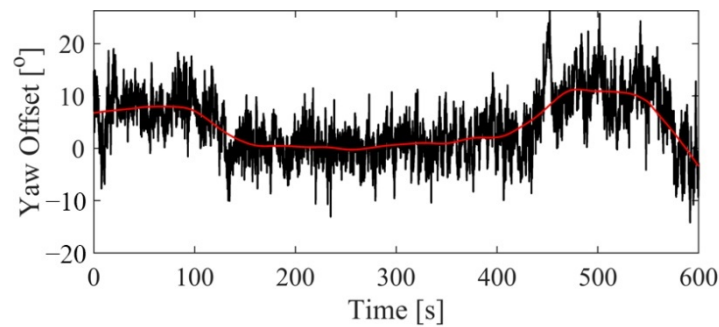
#### 4. Validation of the Computational Model

A numerical simulation of a single wind turbine has been performed to validate the actuator surface model of the SWiFT wind turbine (Vestas V27) [50]. Numerical results are compared against MET data, the turbine supervisory control and acquisition system (SCADA), and SpinnerLidar measurements performed at the SWiFT facility by Sandia National Laboratories and the National Renewable Energy Laboratory [51]. Details of the field experiment and data acquisition can be found in Herges et al. [51].

An initial comparison of the inlet velocity profile obtained from the precursor simulation and the measurements was performed. Measurements of the 10-min time-averaged velocity and turbulence intensity obtained from a MET 2.5D upwind from the wind turbine are compared against the precursor simulation in Figure 3. Although the velocity profile shows good agreement, the turbulence intensity of the precursor simulation shows to be slightly larger than that obtained from the met-tower measurements. Furthermore, the difference in the turbulence intensity seems to be amplified with height, which suggests a slight decrease in turbulent kinetic energy due to atmospheric stratification may have occurred. However, the difference in the turbulence intensity is negligible, around 1% at hub height and 2.7% at 2.15D.

The yaw misalignment of the wind turbine was recorded with the SCADA system. Due to the constantly changing wind direction and the turbine yaw controller not perfectly tracking the wind during the measuring campaign, a time-dependent yaw offset is prescribed to the wind turbine in the simulations to match the measured flow misalignment, shown in Figure 4. In addition, the high-frequency fluctuations of the measured yaw offset, due to the small-scale turbulent coherent structures, are filtered by applying a locally weighted scatterplot smoothing with a window size of 100s to the offset signal. Therefore, a smoother yawing is prescribed to the turbine in the numerical simulation.

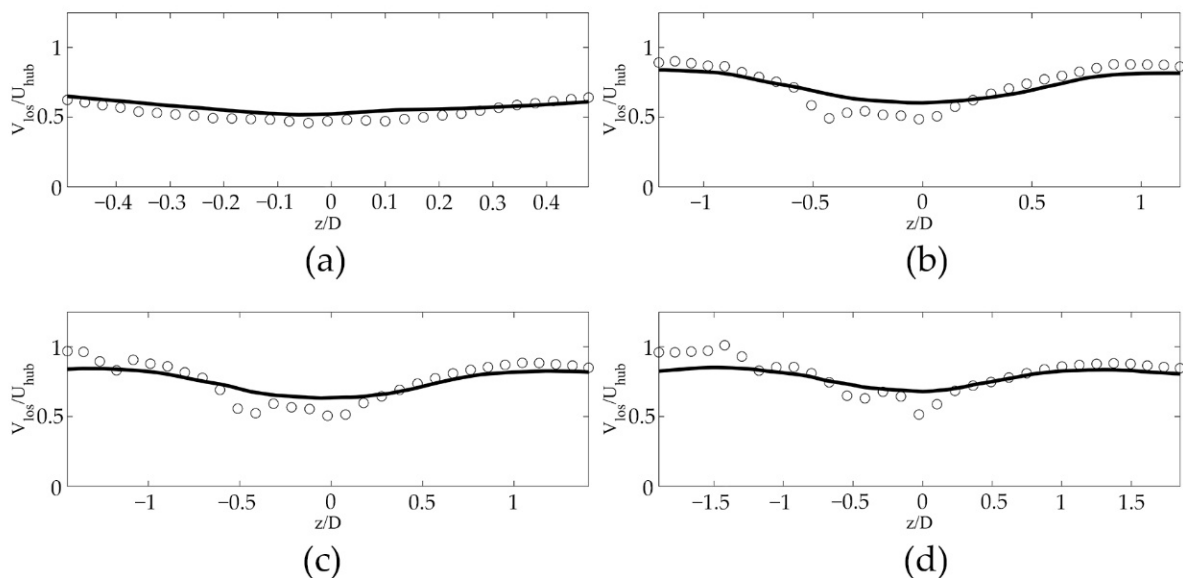




**Figure 4.** Yaw offset in time between the wind and the turbine; (–) SCADA measurement and (–) prescribed to the turbine on the numerical simulation.

The numerical results are compared against the 10-min line-of-sight velocity measurements in the wake of the T1 wind turbine obtained from the nacelle-mounted DTU SpinnerLidar device [51]. The SpinnerLidar performed 984 scans at a constant focus distance from the device every two seconds. In addition, measurements were performed at  $1D$  to  $5D$  behind the wind turbine by varying the focus distance. The SpinnerLidar cycled the focus distance from  $1D$  to  $5D$  every 30 s.

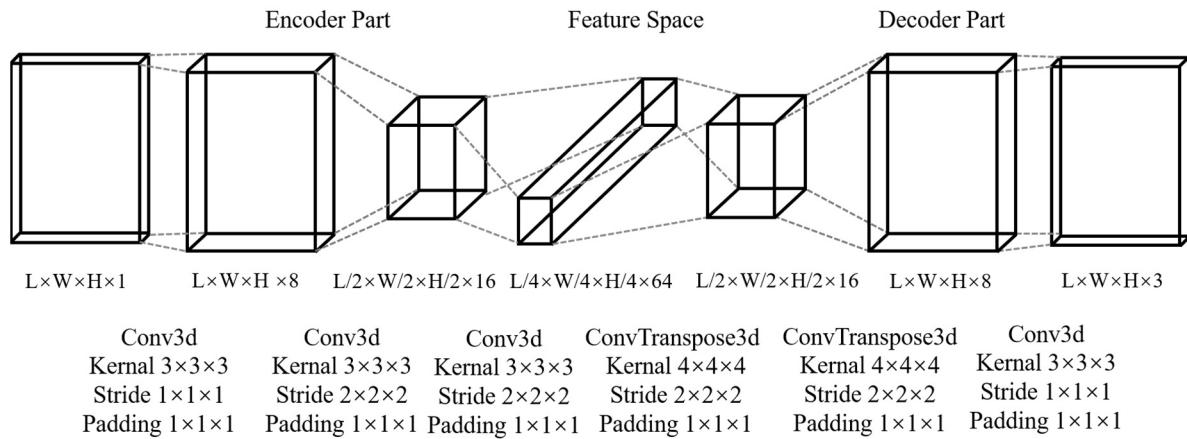
To mimic the SpinnerLidar measurements, the velocity field of the numerical results are decomposed into a line-of-sight velocity ( $V_{LOS}$ ) with vertex, or origin, at the location of the nacelle. For higher fidelity in the comparison to the measurements, the simulated  $V_{LOS}$  field is sampled at the approximate spatial coordinates and time as the SpinnerLidar measurements. The scattered data from the numerical simulation and the SpinnerLidar are interpolated into a spherical surface mesh to compute the 10-min time-averaged  $V_{LOS}$ . A comparison of the horizontal line-of-sight velocity profiles at hub height is shown in Figure 5. Computed velocities in the wake of the turbine are slightly overestimated compared to the SpinnerLidar measurements, specifically close to the center of the rotor. Although the nacelle is modeled and avoids the formation of an unphysical jet in the center of the rotor (commonly observed in standard actuator line models), the momentum deficit seems slightly under-estimated. However, the numerical results from VFS-Wind show good agreement with the SpinnerLidar measurements.



**Figure 5.** Time-averaged line-of-sight velocity: (o) SpinnerLidar and (–) numerical LiDAR from the LES; (a)  $1D$ , (b)  $2D$ , (c)  $3D$ , and (d)  $4D$  behind the wind turbine.

## 5. CNN Autoencoder Model

We employed a CNN autoencoder model to predict the time-averaged flow field by extracting the key flow field features from instantaneous LES results. The CNN algorithm was originally developed to handle image recognition and image classification tasks [52,53]. Therefore, the CNN has some inherent advantages in handling high-dimensional data: it generally consists of convolutional layers and down-sampling layers that can reduce dimensions of the input image and extract abstract features of the image; the weight sharing concept (it will be explained in the next paragraph) used in CNN allows it to handle high-dimensional data using less learnable parameters and avoid location sensitivity problem [52]. As a variant architecture, the CNN autoencoder (Figure 6) consists of an encoder, which extracts features from input data, and a decoder, which is an inverse of the encoder. As a result, such CNN would enable the image reconstruction from the extracted features. Because of its ability to reconstruct field data, the CNN autoencoder has become a popular tool in the field of fluid dynamics, as well [54–57]. For these reasons, we employed the CNN autoencoder model in this study.



**Figure 6.** Schematic of the encoder-decoder CNN. Feature maps are depicted as solid boxes. Convolutional layers are depicted as gray dash lines.  $L \times W \times H \times channels$  represents the dimensions of each feature map.  $L$ ,  $W$ , and  $H$  represent the resolution of the input image in windwise, spanwise and vertical directions, respectively. The layer type, filter size, and stride size of each layer are shown below it. Strides represent the movement step-size of the convolutional filter.

The architecture of the CNN autoencoder used in this work is illustrated in Figure 6. The encoder part includes three 3-dimensional convolution layers for extracting features and down-sample the input data. Each convolution layer includes multiple channels corresponding to different features to be learned. The convolutional layer embedded with a nonlinear activation function and bias operates as follows [55]:

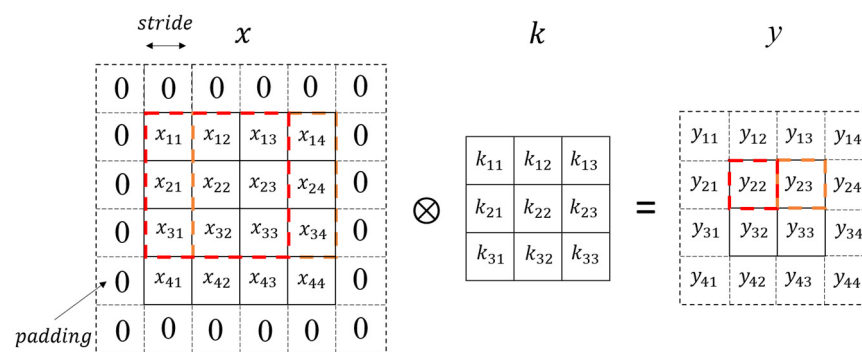
$$q_i^l = \sigma(k_i^l \otimes q^{l-1} + b_i^l), \quad (4)$$

where  $q_i^l$  is the output of  $i$ th channel in the  $l$ th convolutional layer,  $\sigma$  is the rectified linear unit (ReLU) nonlinear activation function [58],  $k_i^l$  is the  $i$ th trainable convolutional filter,  $\otimes$  is the convolution operator,  $q^{l-1}$  is the input of the  $l$ th convolutional layer and  $b_i^l$  is the  $i$ th bias.

In Figure 7, we demonstrate the concept of the convolution operation. In this figure,  $x$  is a  $4 \times 4$  input image with paddings of zero,  $k$  is a  $3 \times 3$  convolution kernel (or filter), while  $y$  is the output of the convolution operation. The convolution window is traversed through the padded input image in both horizontal and vertical directions, where the convolution operation with the filter is performed. The step size of each move is the stride—i.e., the step size from red square to orange square. In the convolution operation, the filter can extract features from the input image, and the learnable weights stay unchanged as the



convolution window moves. As a result, the weights of the filter are shared through the whole input image, and because of the weight sharing only one filter with nine weights is required to extract a feature through the entire padded image—instead of 16 different filters corresponding to each output element. In this approach, the filter will be independent of its location leading to fewer learnable weights and thus a more efficient training process. The convolution operation consists of an inner product between the convolution window (i.e., the red dashed box in the input of Figure 7) and the filter to generate the corresponding output element (e.g., the red dashed box in the output of Figure 7). Since the convolution operation reduces the image size, paddings of zeros are used to control the output size. For instance, the output size of the original  $4 \times 4$  input image in Figure 7 (solid boxes in  $x$ ) is  $2 \times 2$  (solid boxes in  $y$ ). However, with paddings, the output has the same size as the original input. In practice, the convolution operation can work on both 2- and 3-dimensional inputs.



**Figure 7.** Schematic of the convolution operation.  $x$  is the input,  $k$  is the convolution kernel,  $y$  is the output. Zeros around the input are padding. The orange and red squares represent corresponding input and output cells of the first and second convolution operation, respectively.

Since the convolution operation only involves a linear transformation, an activation function is needed to provide the nonlinear transformation into the CNN. Compared to the sigmoid or hyperbolic tangent activation functions, a rectified linear unit (ReLU) in hidden layers can increase the computational efficiency of the machine learning algorithm [58]. The ReLU function is given by [58]

$$\sigma(\theta) = \max(0, \theta), \quad (5)$$

where  $\theta$  is the result of convolution operation plus the linear bias.

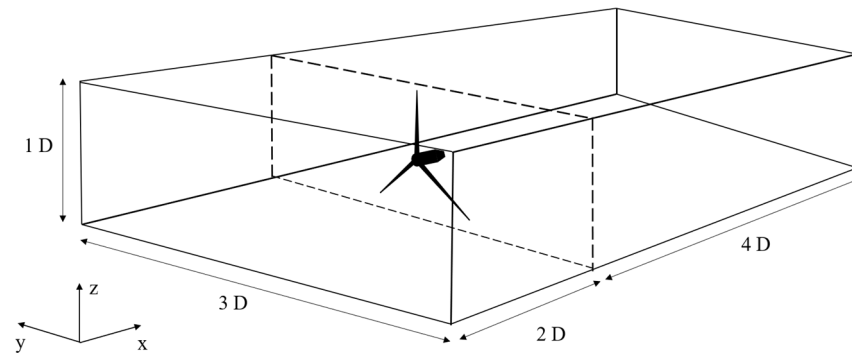
The decoder contains two transpose convolution layers, which are the inverses of the convolution layers, and a convolution layer to up-sample the data and construct the flow field. The discrepancy between output and the target values during the training iterations is calculated using the mean square error (MSE) loss function. Then, the weights in convolutional filters and the biases are updated by the backpropagation algorithm to minimize the loss function [59].

To determine the parameters of the CNN architecture, i.e., the number of layers and channels, and the kernel and stride sizes, a series of parameter combinations are tested to ensure the highest accuracy of results and least number of learnable parameters. The padding sizes are elaborately determined to guarantee the correct output size.

## 6. Results and Discussion

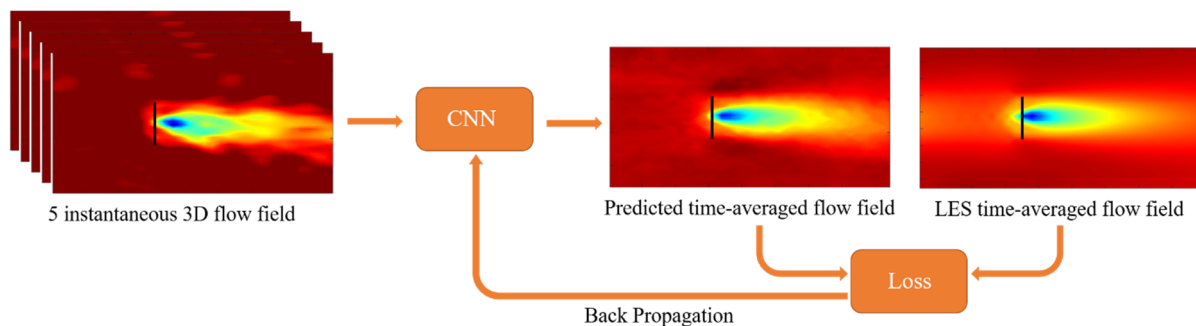
We carried out LES of the SWiFT site under four different wind directions and five different wind velocities to train and validate the predictions of the CNN. First, we carried out the LES for all cases and discarded the initial two flow-through times—i.e., the duration of time it takes for an air particle to travel through the wind farm. Subsequently, the numerical simulations were continued until the first and second-order turbulence statistics

were fully converged. The convergence of the time-averaged flow field is determined using a time-history-analysis approach reported in Khosronejad et al. [60]. During the training process, the fully converged instantaneous flow fields were fed into the CNN at the input layer, while the time-averaged flow field was designated as the target of CNN at the output layer. We note that the samples used in the training and, consequently, validation processes are taken from smaller domains around each turbine. These subdomains are  $6D$  long,  $3D$  wide, and  $1D$  high. The turbine is located  $2D$  downwind from the inlet and centered along the spanwise direction (Figure 8).

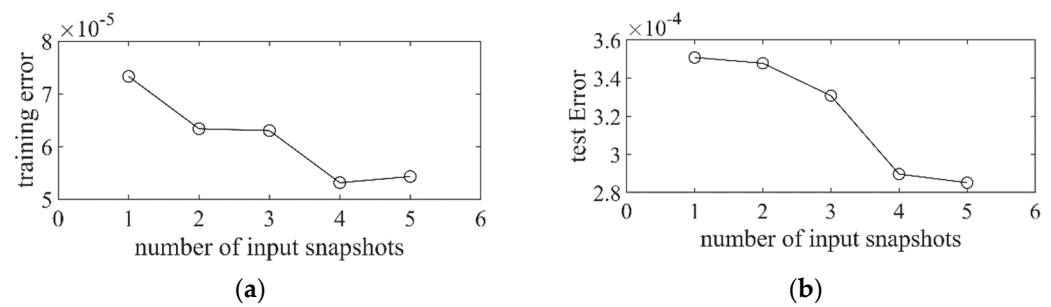


**Figure 8.** Schematic of the local area around turbine of CNNs sub-domain. The dashed line shows the plane of the rotor.

A schematic of the training procedure is shown in Figure 9. The input of each sample is composed of five neighboring snapshots with 1000 time-step intervals to convey the information of flow field fluctuations—induced by the wake meandering and large turbulent structures from the incoming flow. To examine the effect of the number of input snapshots on the accuracy of the trained CNN's predictions (for the time-averaged results), we conducted a systematic analysis in which we used a different number of input snapshots and calculated the computational errors. The computational error refers to the difference between the CNN's and LES results of the training and validation cases. Our findings for the relation between the number of input snapshots and the corresponding computational errors are shown in Figure 10. As seen in this figure, after five snapshots, the computational errors seem to plateau, suggesting that five snapshots are enough to reconstruct the time-averaged flow field. The target value of the CNN at the output layer is a statistically converged time-averaged flow field. Therefore, each training sample consists of five instantaneous snapshots and the time-averaged flow field.



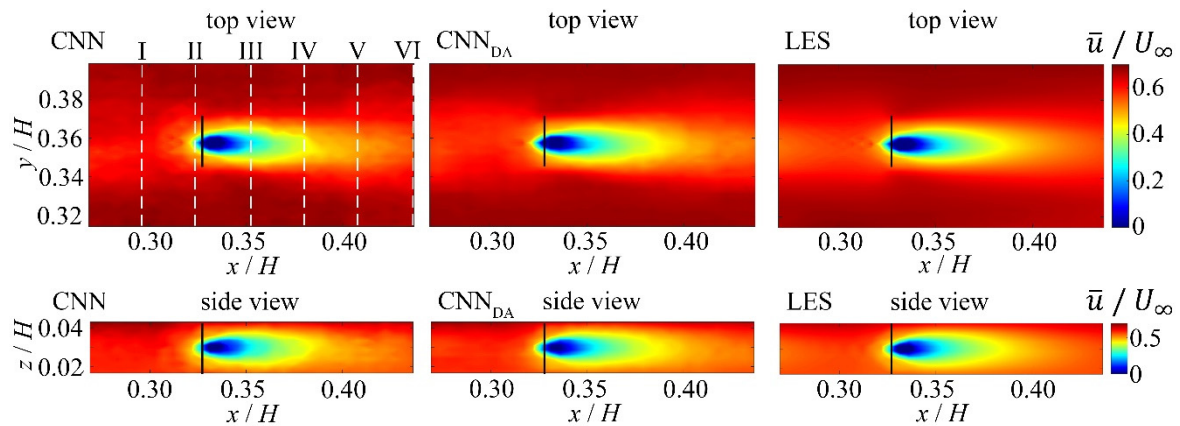
**Figure 9.** Schematic of the training procedure to develop the CNN. The instantaneous velocity fields are fed into the CNN as the input signal while the time-averaged windwise velocity field is enforced as the target of the output signal.



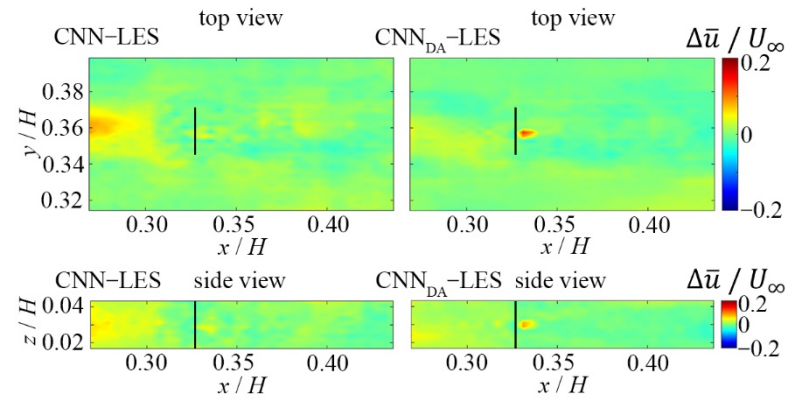
**Figure 10.** Mean square error as a function of the number of input snapshots; (a,b) present the computational errors of the CNN for the training and test dataset, respectively.

The case we considered for the training of the CNN includes the SWiFT site with the north-east wind direction and wind speed of  $U_{\infty} = 7$  m/s. The learning rate of the CNN had an initial value of 0.001 with a decay rate of 0.7 in a step size of 500 training epochs. Overall, 99 samples were used in the training process which took 1500 epochs of iterations until the loss curve was fully converged. The prediction of the training case is compared with the LES result in Figure 11. The difference between CNN prediction and the LES result is presented in Figure 12. Velocity profiles from six spanwise cross-sections are compared in Figure 13. The high accuracy of the CNN prediction demonstrates that the CNN is well trained.

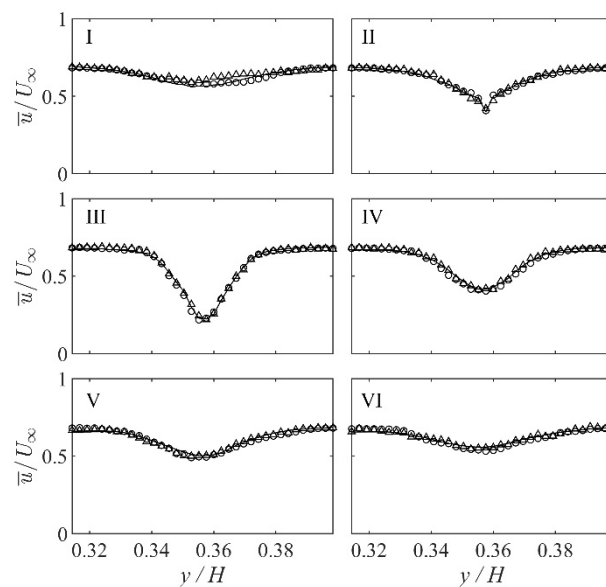
Then the CNN is validated using 19 cases. Similar to the training, five successive snapshots (i.e., five instantaneous flow field data taken from five successive time steps) are time-averaged to produce the limited time-averaged flow field and used as the input to the trained-CNN to generate the reconstructed time-averaged flow field. The CNN-predicted time-averaged flow fields are then compared against the time-averaged results of separately conducted LES. For brevity, we only present the comparison of the overlapped turbine wakes for the four cases, e.g., north-east with  $U_{\infty} = 15$  m/s, south with  $U_{\infty} = 13$  m/s, south-west with  $U_{\infty} = 11$  m/s and west with  $U_{\infty} = 9$  m/s. In Figure 14, we plot the windwise velocity contours from top views at the hub level and side views at the hub layer. In Figure 15, we plot the difference between the CNN predictions and LES results. Additionally, velocity profiles taken from six spanwise cross-sections are compared in Figure 16. In the north-east (Figures 14a, 15a and 16a) and south-west (Figures 14c, 15c and 16c) wind direction cases, the presented turbines are 6D downwind of the other turbines. As seen in these figures, the trained CNN has been able to accurately resemble the velocity deficit in both the upwind and downwind wakes. The largest discrepancy is observed for the cases with the south (Figures 14b, 15b and 16b) and west (Figures 14d, 15d and 16d) wind directions, with the latter showing the maximum discrepancy. The present turbines are 5D and 3D downwind of the other turbines in the south and west case, respectively. The difference in the LES versus the CNN computed velocity fields seems to be caused by the proximity of the wind turbines, i.e., the closer the two turbines, the greater the computational error of the CNN predictions. We note that the CNN results for the velocity fields upwind of the turbines (Figure 15b,d, and profiles I, II in Figure 16b,d) in both the south and west cases seem to be slightly over-estimated. However, the CNN could predict the wake of the centered turbine with great accuracy, suggesting that the relatively higher discrepancy upwind of the turbine is due to the location sensitivity of the trained CNN. Theoretically, because of the “weight sharing” feature of the CNN, a trained CNN should be able to reconstruct the turbine wake at any location within the domain. However, the turbines and their wakes have almost the same location in all the training samples and the upwind wake does not strongly affect the training subdomain. Therefore, the trained CNN deems sensitive to turbine location.



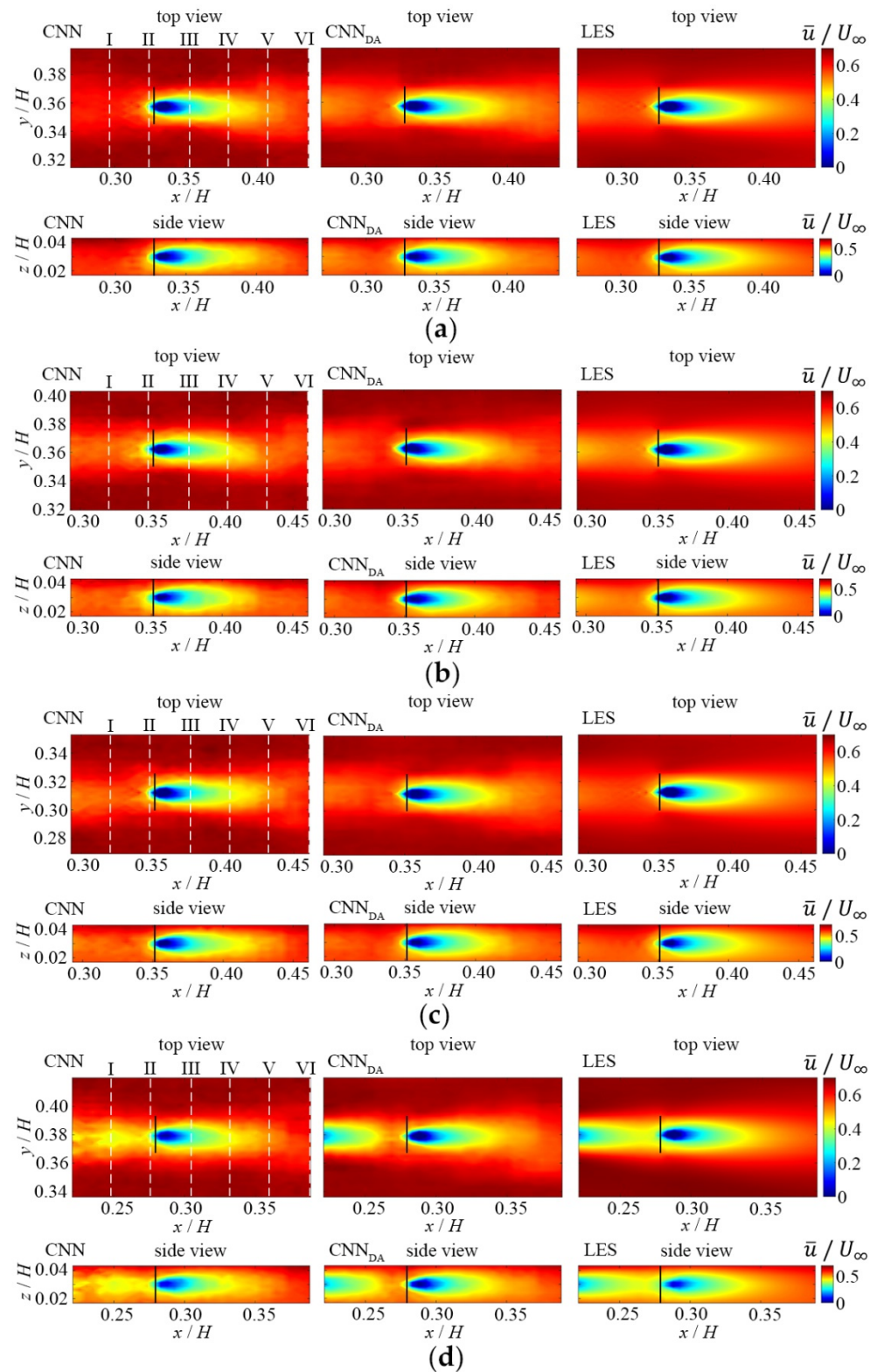
**Figure 11.** Contours of time-averaged velocity normalized with the free-flow velocity for wind turbines in the training case. Top view cross-sections are at hub-height and side-view cross-sections are at the rotor center. Contours are from the CNN, the CNN with data augmentation method ( $CNN_{DA}$ ), and the LES results.



**Figure 12.** Contours of velocity difference between CNN and LES results normalized with the free-flow velocity for wind turbines in the training case. Top view cross-sections are at hub-height and side-view cross-sections are at the rotor center.

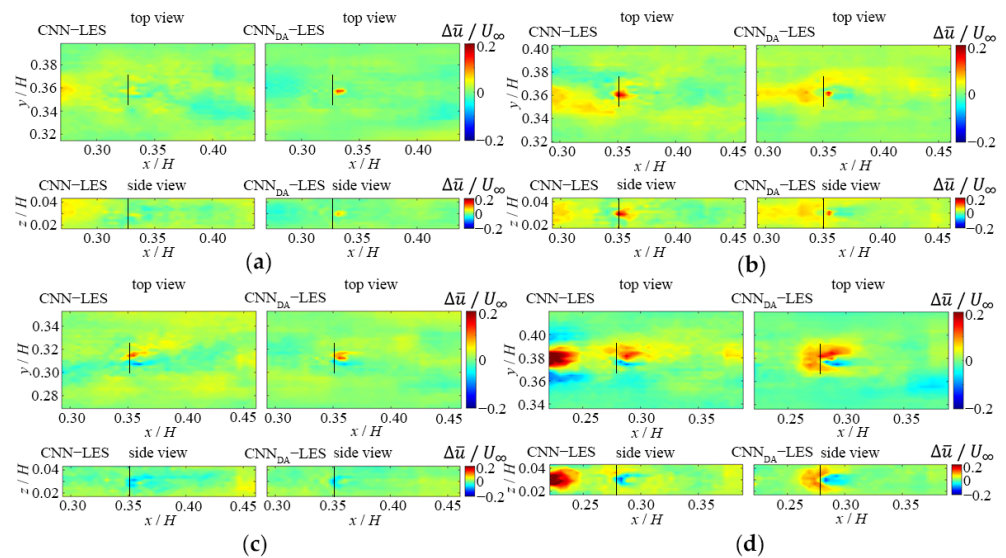


**Figure 13.** Velocity profiles along the spanwise direction at I, II, III, IV, V, and VI in Figure 11. (—) LES, ( $\Delta$ ) CNN, and ( $\circ$ ) CNN with data augmentation.

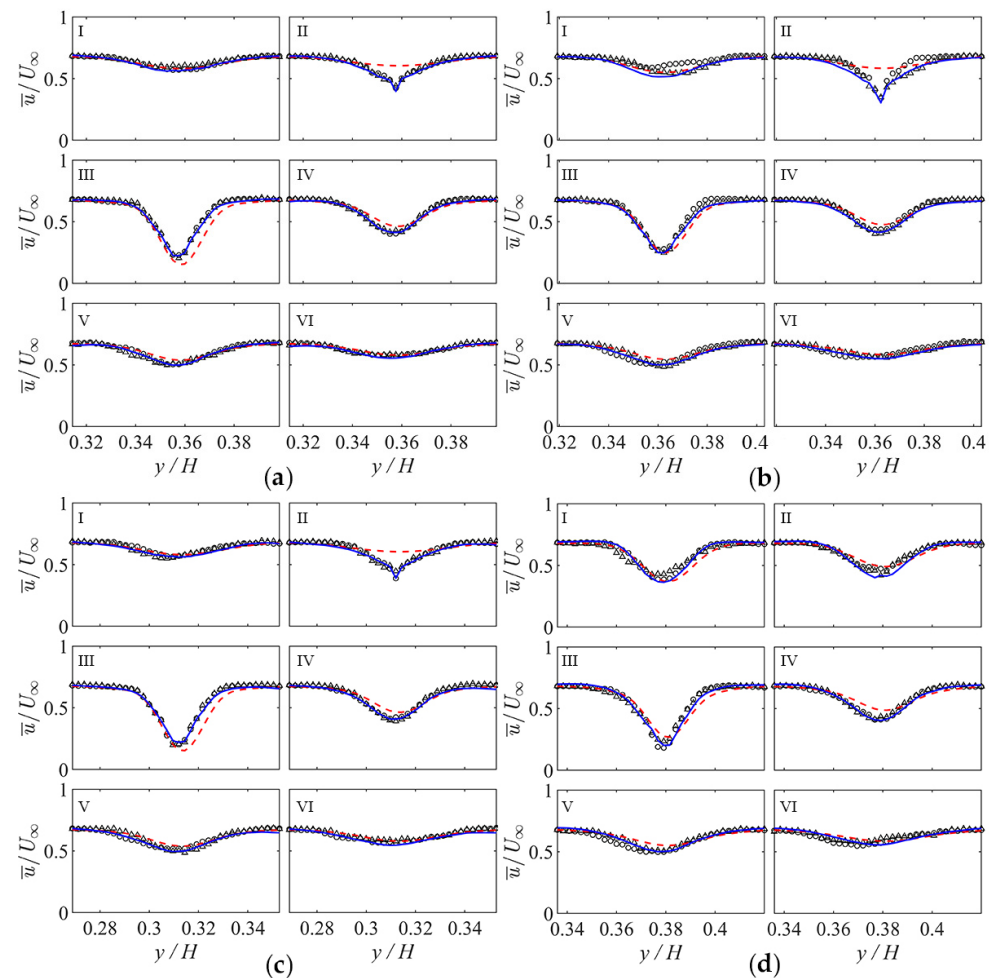


**Figure 14.** Contours of time-averaged velocity normalized with the free-flow velocity for wind turbines. (a) wind turbine 2 in north-east case; (b) wind turbine 3 in south case; (c) wind turbine 3 in south-west case; (d) wind turbine 1 in west case. Top view cross-sections are at hub-height and side-view cross-sections are at the rotor center. Contours are from the CNN, the CNN with data augmentation method (CNN<sub>DA</sub>), and the LES results.





**Figure 15.** Contours of velocity difference between the CNN and LES results normalized with the free-flow velocity for wind turbines. (a) wind turbine 2 in the north-east case; (b) wind turbine 3 in the south case; (c) wind turbine 3 in the south-west case; (d) wind turbine 1 in the west case. Top view cross-sections are at hub-height, and side-view cross-sections are at the rotor center.



**Figure 16.** Velocity profiles along the spanwise direction at I, II, III, IV, V, and VI in Figure 14. (a) wind turbine 2 in the north-east case; (b) wind turbine 3 in the south case; (c) wind turbine 3 in the south-west case; (d) wind turbine 1 in the west case. (—) LES, (---) Analytical model, ( $\Delta$ ) CNN, and ( $\circ$ ) CNN with data augmentation.



To overcome the location sensitivity problem, we used the data augmentation technique. In this approach, instead of having a fixed location, the subdomains around the turbines are randomly moved up to 5D upwind. We also increase the size of training samples by a factor of five. Using this approach, the training samples contain the turbines and their wakes which are randomly scattered in each subdomain. The training and validation results of the CNN trained using the data augmentation technique (CNN<sub>DA</sub>) are depicted in Figures 11–16. The velocity deficits in the upwind wakes of the south and west cases are predicted more accurately in comparison to the results without data augmentation (Figure 15b,d, and I in Figure 16b,d). However, the discrepancies of the velocities around the turbines are still large in Figure 15b,d, because in these cases, the interaction with upwind turbine's wake is stronger than the training case.

The analytical model developed by Bastankhah and Porté-Agel [9] is used to compare against the CNN model. In this analytical model, the velocity deficit in the wake of a turbine is described as [19]:

$$\frac{\Delta u}{u_0} = \left( 1 - \sqrt{1 - \frac{C_T}{8 \left( \frac{\sigma_y \sigma_z}{D^2} \right)}} \right) \exp \left( -0.5 \left[ \left( \frac{y}{\sigma_y} \right)^2 + \left( \frac{z}{\sigma_z} \right)^2 \right] \right), \quad (6)$$

where the  $\Delta u$  is the velocity deficit in the wake,  $u_0$  is the mean wind velocity perceived by the wind turbine,  $C_T$  is thrust coefficient,  $\sigma_y$  and  $\sigma_z$  are the wake widths in spanwise and vertical directions, respectively, which are given by [19]:

$$\frac{\sigma_y}{D} = k_y \frac{x}{D} + 0.2 \sqrt{\frac{1 + \sqrt{1 - C_T}}{2\sqrt{1 - C_T}}}, \quad (7)$$

$$\frac{\sigma_z}{D} = k_z \frac{x}{D} + 0.2 \sqrt{\frac{1 + \sqrt{1 - C_T}}{2\sqrt{1 - C_T}}}, \quad (8)$$

where  $k_y$  and  $k_z$  are the wake growth rates in spanwise and vertical directions, respectively. In the training and validation cases,  $C_T = 0.8$ . The wake growth rates  $k_y = 0.075$  and  $k_z = 0.075$  are obtained by fitting the training case. For the turbines in the wake, a wake superposition model is considered [61]:

$$u = U_{hub} - \sqrt{\sum_i^n \Delta u_i^2}, \quad (9)$$

where  $U_{hub}$  is the free stream velocity at the hub height,  $u$  is the velocity in the wake of current turbine,  $\Delta u_i$  is the wake velocity deficit of the  $i$ th turbine in stand-alone condition,  $n$  is the number of superposition wakes. The velocity profiles of the analytical model are presented in Figure 16. The analytical model has a good performance in far wakes. However, the velocity deficits are overpredicted in the near wake areas (III in Figure 16) and underpredicted in the near upwind areas (II in Figure 16). In comparison, the presented CNN<sub>DA</sub> model seems to have a better performance than the analytical model in predicting the velocity field.

The discrepancy between the CNN<sub>DA</sub> predictions and the LES time-averaged results were quantified using the coefficient of determination ( $R^2$ ), mean absolute error (MAE), root mean square error (RMSE), and the mean absolute relative error (MARE). These statistical error indices are defined as follows [62]:

$$R^2 = 1 - \frac{\sum_{i=1}^N \left( \psi_{i(CNN)} - \psi_{i(LES)} \right)^2}{\sum_{i=1}^N \left( \psi_{i(CNN)} - \bar{\psi}_{i(CNN)} \right)^2}, \quad (10)$$

$$\text{MAE} = \frac{\sum_{i=1}^N |\psi_{i(\text{CNN})} - \psi_{i(\text{LES})}|}{N}, \quad (11)$$

$$\text{RMSE} = \left( \frac{\sum_{i=1}^N (\psi_{i(\text{CNN})} - \psi_{i(\text{LES})})^2}{N} \right)^{0.5}, \quad (12)$$

$$\text{MARE} = \frac{1}{N} \sum_{i=1}^N \frac{|\psi_{i(\text{CNN})} - \psi_{i(\text{LES})}|}{\psi_{i(\text{LES})}}, \quad (13)$$

where  $\psi_{i(\text{CNN})}$  is the predicted value (of time-averaged windwise velocity component) using the CNN<sub>DA</sub>,  $\psi_{i(\text{LES})}$  is the value obtained from the LES model,  $\bar{\psi}_{i(\text{CNN})}$  is the mean predicted value using the CNN<sub>DA</sub>, and  $N$  is the total number of samples, i.e., the total number of computational nodes in the subdomain surrounding the turbine.

The discrepancies between the CNN<sub>DA</sub> predictions and the LES time-averaged results for the four cases are presented in Table 2. The CNN<sub>DA</sub> predictions maintain high accuracy for the turbines located in the free flow for all wind conditions (i.e., various magnitudes and directions). The largest discrepancies are observed for the turbines located in the wake and, specifically, the west wind direction case, where the distance between the turbines is the smallest. Nevertheless, the  $R^2$  for all cases are over 0.95 and the RMSE less than 3%, which is quite remarkable.

**Table 2.** Statistical error indices of the CNN<sub>DA</sub> relative to the LES results for different wind direction cases.

Wind Direction	Turbine	$R^2$	MAE	RMSE	MARE
NE	T <sub>1</sub>	0.99	0.0055	0.0097	0.01
	T <sub>2</sub>	0.99	0.0066	0.0111	0.01
	T <sub>3</sub>	0.99	0.0045	0.0080	0.01
S	T <sub>1</sub>	0.98	0.0083	0.0147	0.02
	T <sub>2</sub>	0.98	0.0080	0.0162	0.01
	T <sub>3</sub>	0.97	0.0106	0.0149	0.02
SW	T <sub>1</sub>	0.97	0.0102	0.0095	0.02
	T <sub>2</sub>	0.98	0.0074	0.0143	0.01
	T <sub>3</sub>	0.92	0.0166	0.0201	0.03
W	T <sub>1</sub>	0.95	0.0140	0.0256	0.03
	T <sub>2</sub>	0.95	0.0138	0.0267	0.03
	T <sub>3</sub>	0.96	0.0116	0.0181	0.02

The computational efficiency of the proposed CNN model has a great advantage over the LES. For each case, the LES required over  $8 \times 10^4$  CPU hours to generate the time-averaged flow fields. In comparison, although the proposed CNN model requires about 10 CPU hours for training, the well-trained model only requires 50 s to reconstruct the time-averaged flow field. Considering the LES requires  $9.6 \times 10^3$  to generate the instantaneous flow fields for inputs, the total cost of the proposed CNN model is still 88% less than the LES.

Now we turn our attention to the possibility of using the proposed CNN to predict aerodynamic power production of the individual turbines using the predicted velocity fields as follows:

$$P_{aero} = \frac{1}{2} \rho A u^3, \quad (14)$$

where  $P_{aero}$  is the aerodynamic power production,  $\rho = 1.225 \text{ kg m}^{-3}$  is the air density,  $A$  is the frontal rotor area, and  $u$  is the mean time-averaged windwise velocity component over the rotor area. In Figure 17, we compare the aerodynamic power productions of the four turbines using the LES results and CNN<sub>DA</sub> predictions. Wind velocities of the four

cases range from 7 m/s to 15 m/s. Even though the velocity field discrepancies between LES and CNN<sub>DA</sub> are satisfactory, all wind turbines across all wind velocities overestimated the predicted power productions. As expected, the largest discrepancy is at turbine 1 in the west wind direction case, where the velocity deficit around the rotor area is underestimated (for instance, II in Figure 16d).

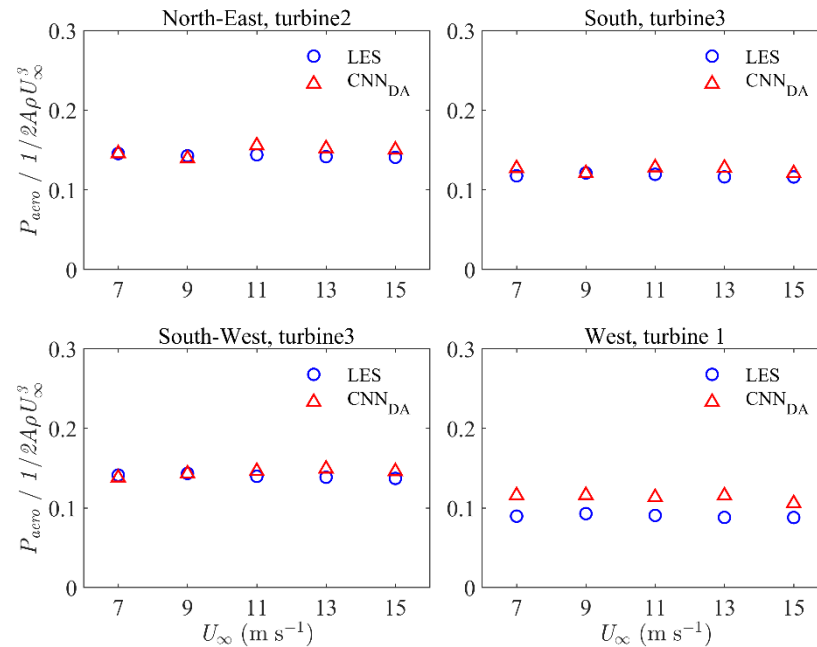


Figure 17. Power production of turbines located in the wake.

## 7. Conclusions

In this study, we examined the capability of the CNN autoencoder to reconstruct the time-averaged flow field around the wind turbines at the SWiFT facility and predict turbine power output. LES of the SWiFT facility with four different wind conditions (i.e., north-east, south, south-west, and west directions with wind speeds of 7, 9, 11, 13, and 15 m/s) were performed to generate training and validation data for the CNN. A six-layer CNN autoencoder was developed and trained using both instantaneous and time-averaged LES results around three individual turbines using the north-east wind direction case. The input of every sample is constructed using five instantaneous velocity fields to reflect the temporal variations of turbulent structures. Subsequently, the trained CNN is validated and compared with time-averaged results of the additional large-eddy simulations. Based on the findings of this study, the following conclusions can be drawn:

- (1) The trained CNN can successfully predict the time-averaged flow field around individual turbines, while the data augmentation technique can effectively address the location sensitivity of the trained CNN. The predicted flow field clearly reflects the main features of the turbine wakes obtained from LES. The velocity profiles drawn from CNN predictions agree well with LES time-averaged results and the overall relative errors are no more than 3%. The presented model has a good generality in different wind speeds. However, wake overlapping will affect the accuracy of the predictions. Different turbine distances lead to different wake-turbine interaction effects and, thus, the flow structure near the turbine varies significantly. Since the CNN model is only trained using one turbine wake interaction case, the generality in different wake interaction cases is appreciable. In a future study, we will consider more wake overlapping cases in the training dataset to enable better flow field predictions.
- (2) The computational cost associated with the LES to generate the time-average flow field for each case was over  $8 \times 10^4$  CPU hours, whereas the CNN required nearly 50 s to reconstruct the same flow field. Considering the training cost of about 10 CPU hours

and the cost of LES to produce instantaneous flow field (i.e.,  $9.6 \times 10^3$  CPU hours) for the inputs of the CNN, the total cost of the proposed CNN is 88% less than that of the LES. Therefore, the proposed CNN algorithms could enable reliable predictions of the wake flow field at a fraction of the cost required by the LES.

- (3) The CNN predictions for the aerodynamic power productions were in good agreement with the LES results, except for the turbine located in the near wake of the upwind turbine owing to an underestimation of the velocity deficit within the wake. Overall, the comparisons between the LES results and CNN predictions of the SWiFT wind turbines demonstrate the potential of the developed CNN autoencoder for predicting time-averaged flow fields and the power production of wind turbines while being several orders of magnitude less computationally expensive than high-fidelity numerical simulations.

**Author Contributions:** Conceptualization, Z.Z., A.K.; methodology, Z.Z., A.K., T.H.; software, A.K., F.S.; validation, C.S., T.H.; formal analysis, Z.Z., A.K., C.S., F.S.; investigation, Z.Z., A.K., C.S.; resources, A.K., F.S.; data curation, C.S.; writing—original draft preparation, Z.Z., C.S.; writing—review and editing, A.K., F.S., T.H.; visualization, Z.Z., C.S.; supervision, A.K.; project administration, A.K., F.S.; funding acquisition, A.K., F.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the National Offshore Wind Research and Development Consortium (NOWRDC) under agreement number 147503.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the size of the dataset.

**Acknowledgments:** The computational resources were provided by the Civil Engineering Department, Stony Brook Research Computing and Cyberinfrastructure, and the Institute for Advanced Computational Science at Stony Brook University. Additional computational resources were provided by the Minnesota Supercomputing Institute at the University of Minnesota. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Thomsen, K.; Sørensen, P. Fatigue loads for wind turbines operating in wakes. *J. Wind Eng. Ind. Aerodyn.* **1999**, *80*, 121–136. [\[CrossRef\]](#)
- El-Asha, S.; Zhan, L.; Iungo, G.V. Quantification of power losses due to wind turbine wake interactions through SCADA, meteorological and wind LiDAR data. *Wind Energy* **2017**, *20*, 1823–1839. [\[CrossRef\]](#)
- Blondel, F.; Cathelain, M. An alternative form of the super-Gaussian wind turbine wake model. *Wind Energy Sci.* **2020**, *5*, 1225–1236. [\[CrossRef\]](#)
- 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* **2018**, *233–234*, 975–984. [\[CrossRef\]](#)
- Regodeseves, P.G.; Morros, C.S. Unsteady numerical investigation of the full geometry of a horizontal axis wind turbine: Flow through the rotor and wake. *Energy* **2020**, *202*, 117674. [\[CrossRef\]](#)
- Hornshøj-Møller, S.D.; Nielsen, P.D.; Forooghi, P.; Abkar, M. Quantifying structural uncertainties in Reynolds-averaged Navier–Stokes simulations of wind turbine wakes. *Renew. Energy* **2020**, *164*, 1550–1558. [\[CrossRef\]](#)
- Jensen, N.O. *A Note on Wind Generator Interaction*; Risø National Laboratory: Roskilde, Denmark, 1983.
- Hamed, R.; Javaheri, A.; Dehghan, O.; Torabi, F. A semi-analytical model for velocity profile at wind turbine wake using blade element momentum. *Energy Equip. Syst.* **2015**, *3*, 13–24. [\[CrossRef\]](#)
- Bastankhah, M.; Porté-Agel, F. A new analytical model for wind-turbine wakes. *Renew. Energy* **2014**, *70*, 116–123. [\[CrossRef\]](#)

10. Gao, X.; Yang, H.; Lu, L. Optimization of wind turbine layout position in a wind farm using a newly-developed two-dimensional wake model. *Appl. Energy* **2016**, *174*, 192–200. [\[CrossRef\]](#)
11. Keane, A.; Aguirre, P.E.O.; Ferchland, H.; Clive, P.; Gallacher, D. An analytical model for a full wind turbine wake. *J. Phys. Conf. Ser.* **2016**, *753*, 032039. [\[CrossRef\]](#)
12. Keane, A. Advancement of an analytical double-Gaussian full wind turbine wake model. *Renew. Energy* **2021**, *171*, 687–708. [\[CrossRef\]](#)
13. Sun, H.; Yang, H. Study on an innovative three-dimensional wind turbine wake model. *Appl. Energy* **2018**, *226*, 483–493. [\[CrossRef\]](#)
14. Sun, H.; Yang, H. Numerical investigation of the average wind speed of a single wind turbine and development of a novel three-dimensional multiple wind turbine wake model. *Renew. Energy* **2019**, *147*, 192–203. [\[CrossRef\]](#)
15. Ishihara, T.; Qian, G.-W. A new Gaussian-based analytical wake model for wind turbines considering ambient turbulence intensities and thrust coefficient effects. *J. Wind Eng. Ind. Aerodyn.* **2018**, *177*, 275–292. [\[CrossRef\]](#)
16. Qian, G.-W.; Ishihara, T. A New Analytical Wake Model for Yawed Wind Turbines. *Energies* **2018**, *11*, 665. [\[CrossRef\]](#)
17. Cheng, Y.; Zhang, M.; Zhang, Z.; Xu, J. A new analytical model for wind turbine wakes based on Monin-Obukhov similarity theory. *Appl. Energy* **2019**, *239*, 96–106. [\[CrossRef\]](#)
18. Lin, M.; Porté-Agel, F. Large-Eddy Simulation of Yawed Wind-Turbine Wakes: Comparisons with Wind Tunnel Measurements and Analytical Wake Models. *Energies* **2019**, *12*, 4574. [\[CrossRef\]](#)
19. Porté-Agel, F.; Bastankhah, M.; Shamsoddin, S. Wind-Turbine and Wind-Farm Flows: A Review. *Bound.-Layer Meteorol.* **2019**, *174*, 1–59. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Fuertes, F.C.; Markfort, C.D.; Porté-Agel, F. Wind Turbine Wake Characterization with Nacelle-Mounted Wind Lidars for Analytical Wake Model Validation. *Remote Sens.* **2018**, *10*, 668. [\[CrossRef\]](#)
21. Archer, C.L.; Vassel-Bé-Hagh, A.; Yan, C.; Wu, S.; Pan, Y.; Brodie, J.F.; Maguire, A.E. Review and evaluation of wake loss models for wind energy applications. *Appl. Energy* **2018**, *226*, 1187–1207. [\[CrossRef\]](#)
22. Foti, D.; Yang, X.; Guala, M.; Sotiropoulos, F. Wake meandering statistics of a model wind turbine: Insights gained by large eddy simulations. *Phys. Rev. Fluids* **2016**, *1*, 044407. [\[CrossRef\]](#)
23. Foti, D.; Yang, X.; Sotiropoulos, F. Similarity of wake meandering for different wind turbine designs for different scales. *J. Fluid Mech.* **2018**, *842*, 5–25. [\[CrossRef\]](#)
24. Foti, D.; Yang, X.; Campagnolo, F.; Maniaci, D.; Sotiropoulos, F. Wake meandering of a model wind turbine operating in two different regimes. *Phys. Rev. Fluids* **2018**, *3*, 054607. [\[CrossRef\]](#)
25. Xie, S.; Archer, C.L. A Numerical Study of Wind-Turbine Wakes for Three Atmospheric Stability Conditions. *Bound.-Layer Meteorol.* **2017**, *165*, 87–112. [\[CrossRef\]](#)
26. Yang, X.; Pakula, M.; Sotiropoulos, F. Large-eddy simulation of a utility-scale wind farm in complex terrain. *Appl. Energy* **2018**, *229*, 767–777. [\[CrossRef\]](#)
27. Liu, Z.; Lu, S.; Ishihara, T. Large eddy simulations of wind turbine wakes in typical complex topographies. *Wind Energy* **2021**, *24*, 857–886. [\[CrossRef\]](#)
28. Qian, G.-W.; Ishihara, T. Numerical study of wind turbine wakes over escarpments by a modified delayed detached eddy simulation. *J. Wind Eng. Ind. Aerodyn.* **2019**, *191*, 41–53. [\[CrossRef\]](#)
29. Foti, D.; Yang, X.; Shen, L.; Sotiropoulos, F. Effect of wind turbine nacelle on turbine wake dynamics in large wind farms. *J. Fluid Mech.* **2019**, *869*, 1–26. [\[CrossRef\]](#)
30. Santoni, C.; Carrasquillo, K.; Arenas-Navarro, I.; Leonardi, S. Effect of tower and nacelle on the flow past a wind turbine. *Wind Energy* **2017**, *20*, 1927–1939. [\[CrossRef\]](#)
31. Sedaghatzadeh, N.; Arjomandi, M.; Kelso, R.; Cazzolato, B.; Ghayesh, M.H. Modelling of wind turbine wake using large eddy simulation. *Renew. Energy* **2018**, *115*, 1166–1176. [\[CrossRef\]](#)
32. Japar, F.; Mathew, S.; Narayanaswamy, B.; Lim, C.M.; Hazra, J. Estimating the wake losses in large wind farms: A machine learning approach. In Proceedings of the Innovative Smart Grid Technologies Conference 2014, Washington, DC, USA, 19–22 February 2014; pp. 1–5. [\[CrossRef\]](#)
33. Sun, H.; Qiu, C.; Lu, L.; Gao, X.; Chen, J.; Yang, H. Wind turbine power modelling and optimization using artificial neural network with wind field experimental data. *Appl. Energy* **2020**, *280*, 115880. [\[CrossRef\]](#)
34. Wilson, B.; Wakes, S.; Mayo, M. Surrogate modeling a computational fluid dynamics-based wind turbine wake simulation using machine learning. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–8. [\[CrossRef\]](#)
35. Ti, Z.; Deng, X.W.; Yang, H. Wake modeling of wind turbines using machine learning. *Appl. Energy* **2019**, *257*, 114025. [\[CrossRef\]](#)
36. Ti, Z.; Deng, X.W.; Zhang, M. Artificial Neural Networks based wake model for power prediction of wind farm. *Renew. Energy* **2021**, *172*, 618–631. [\[CrossRef\]](#)
37. Yang, X. Towards the development of a wake meandering model based on neural networks. *J. Phys. Conf. Ser.* **2020**, *1618*, 062026. [\[CrossRef\]](#)
38. Zhang, J.; Zhao, X. Machine-Learning-Based Surrogate Modeling of Aerodynamic Flow Around Distributed Structures. *AIAA J.* **2021**, *59*, 868–879. [\[CrossRef\]](#)
39. King, R.N.; Adcock, C.; Annoni, J.; Dykes, K. Data-Driven Machine Learning for Wind Plant Flow Modeling. *J. Phys. Conf. Ser.* **2018**, *1037*, 072004. [\[CrossRef\]](#)



40. Ali, N.; Calaf, M.; Cal, R.B. Clustering sparse sensor placement identification and deep learning based forecasting for wind turbine wakes. *J. Renew. Sustain. Energy* **2021**, *13*, 023307. [CrossRef]
41. Renganathan, S.A.; Maulik, R.; Letizia, S.; Iungo, G.V. Data-Driven Wind Turbine Wake Modeling via Probabilistic Machine Learning. 2021. Available online: <http://arxiv.org/abs/2109.02411> (accessed on 9 October 2021).
42. Aird, J.A.; Quon, E.W.; Barthelmie, R.J.; Debnath, M.; Doubrawa, P.; Pryor, S.C. Region-Based Convolutional Neural Network for Wind Turbine Wake Characterization in Complex Terrain. *Remote Sens.* **2021**, *13*, 4438. [CrossRef]
43. Khosronejad, A.; Sotiropoulos, F. A short note on the simulation of turbulent stratified flow and mobile bed interaction using the continuum coupled flow and morphodynamics model. *Environ. Fluid Mech.* **2020**, *20*, 1511–1525. [CrossRef]
44. Germano, M.; Piomelli, U.; Moin, P.; Cabot, W.H. A dynamic subgrid-scale eddy viscosity model. *Phys. Fluids A Fluid Dyn.* **1991**, *3*, 1760–1765. [CrossRef]
45. Gilmanov, A.; Sotiropoulos, F. A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *J. Comput. Phys.* **2005**, *207*, 457–492. [CrossRef]
46. Kang, S.; Lightbody, A.; Hill, C.; Sotiropoulos, F. High-resolution numerical simulation of turbulence in natural waterways. *Adv. Water Resour.* **2011**, *34*, 98–113. [CrossRef]
47. Yang, X.; Sotiropoulos, F. A new class of actuator surface models for wind turbines. *Wind Energy* **2018**, *21*, 285–302. [CrossRef]
48. Yang, X.; Zhang, X.; Li, Z.; He, G.-W. A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. *J. Comput. Phys.* **2009**, *228*, 7821–7836. [CrossRef]
49. Berg, J.; Bryant, J.; LeBlanc, B.; Maniaci, D.C.; Naughton, B.; Paquette, J.A.; Resor, B.R.; White, J.; Kroeker, D. Scaled Wind Farm Technology Facility Overview. In Proceedings of the 32nd ASME Wind Energy Symposium, National Harbor, MD, USA, 13–17 January 2014. [CrossRef]
50. Kelley, C.L.; Ennis, B.L. *SWiFT Site Atmospheric Characterization*; Technical Report; Sandia National Lab.: Albuquerque, NM, USA, 2016.
51. Herges, T.G.; Keyantuo, P. Robust Lidar Data Processing and Quality Control Methods Developed for the SWiFT Wake Steering Experiment. *J. Phys. Conf. Ser.* **2019**, *1256*, 012005. [CrossRef]
52. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
53. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2012**, *25*, 1097–1105. [CrossRef]
54. Guo, X.; Li, W.; Iorio, F. Convolutional Neural Networks for Steady Flow Approximation. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 481–490. [CrossRef]
55. Sekar, V.; Jiang, Q.; Shu, C.; Khoo, B.C. Fast flow field prediction over airfoils using deep learning approach. *Phys. Fluids* **2019**, *31*, 057103. [CrossRef]
56. Morimoto, M.; Fukami, K.; Zhang, K.; Nair, A.G.; Fukagata, K. Convolutional Neural Networks for Fluid Flow Analysis: Toward Effective Metamodeling and Low-Dimensionalization. 2021. Available online: <http://arxiv.org/abs/2101.02535> (accessed on 20 February 2021).
57. Nakamura, T.; Fukami, K.; Hasegawa, K.; Nabae, Y.; Fukagata, K. Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow. *Phys. Fluids* **2021**, *33*, 025116. [CrossRef]
58. Glorot, X.; Bordes, A.; Bengio, Y. Deep sparse rectifier neural networks. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 315–323.
59. Hecht-Nielsen, R. Theory of the backpropagation neural network. In *Neural Networks for Perception*; Academic Press: Cambridge, MA, USA, 1992; pp. 65–93.
60. Khosronejad, A.; Flora, K.; Kang, S. Effect of Inlet Turbulent Boundary Conditions on Scour Predictions of Coupled LES and Morphodynamics in a Field-Scale River: Bankfull Flow Conditions. *J. Hydraul. Eng.* **2020**, *146*, 04020020. [CrossRef]
61. Voutsinas, S.; Rados, K.; Zervos, A. On the analysis of wake effects in wind parks. *Wind Eng.* **1990**, *14*, 204–219.
62. Ebtehaj, I.; Bonakdari, H. A comparative study of extreme learning machines and support vector machines in prediction of sediment transport in open channels. *Int. J. Eng. Trans. B Appl.* **2016**, *29*, 1499–1506.