Journal of Physics: Conference Series

PAPER • OPEN ACCESS

WakeNet 0.1 - A Simple Three-dimensional Wake Model Based on Convolutional Neural Networks

To cite this article: Henrik Asmuth and Henry Korb 2022 *J. Phys.: Conf. Ser.* **2265** 022066

View the article online for updates and enhancements.

You may also like

- Numerical analysis of flow interaction of turbine system in two-stage turbocharger of internal combustion engine
  Y B Liu, W L Zhuge, Y J Zhang et al.
- Proper Orthogonal Decomposition (POD) of the Wake Flow Field of a Model Wind Turbine and a Porous Disc under Different Freestream Turbulence Intensity Conditions
  Burahan Öztürk, Abdelrahman Hassanein, M Turul Akpolat et al.
- Large-Eddy Simulations of wind turbine wakes in sheared inflows
  Anand Parinam, Pierre Benard, Dominic von Terzi et al.

This content was downloaded from IP address 130.238.172.43 on 04/09/2023 at 13:22

WakeNet 0.1 - A Simple Three-dimensional Wake Model Based on Convolutional Neural Networks

Henrik Asmuth[1] **and Henry Korb**[1]

[1] Uppsala University, Wind Energy Section, Campus Gotland, 621 57 Visby, Sweden

E-mail: henrik.asmuth@geo.uu.se

Abstract. Deep convolutional neural networks are a promising machine learning approach for computationally efficient predictions of flow fields. In this work we present a simple modelling framework for the prediction of the time-averaged three-dimensional flow field of wind turbine wakes. The proposed model requires the mean inflow upstream of the turbine, aerodynamic data of the turbine and the tip-speed ratio as input data. The output comprises all three mean velocity components as well as the turbulence intensity. The model is trained with the flow statistics of 900 actuator line large-eddy simulations of a single turbine in various inflow and operating conditions. The model is found to accurately predict the characteristic features of the wake flow. The overall accuracy and efficiency of the model render it as a promising approach for future wind turbine wake predictions.

1. Introduction

The modelling of wind turbine wakes and their interaction with the atmospheric boundary layer (ABL) continues to be one of the greatest challenges in wind energy research today [1, 2]. While non-linear numerical approaches such as RANS (Reynolds-averaged Navier Stokes) or LES (Large-eddy Simulation) can provide accurate predictions of the occurring flow phenomena, they often remain too computationally expensive for wide-spread applications in the industrial practice. On the other hand, more efficient engineering models often hinge on empirical tuning factors and tend to fail in off-calibration situations. In recent years, deep learning techniques have given rise to various novel approaches for efficient fluid dynamics modelling. This includes topics such as turbulence [3, 4] and wall modelling [5, 6] as well as surrogate models for entire 2D or 3D flow fields for case-specific problems [7, 8, 9]. One promising approach for computationally efficient flow field predictions are convolutional neural networks (CNNs). CNNs are based on filter kernels (convolutions) mapping the information of input data with a grid-like topology to multiple feature maps. Feeding the stacked feature maps of one convolutional layer to the next allows for a progressively more complex encoding of larger features. When compared to classical fully-connected neural networks, CNNs comprise notably fewer tunable parameters, since the kernel size of each layer is typically a lot smaller than the size of the input data [10, 11]. Therefore, CNNs have proven to be particularly suitable to process large multi-dimensional data such images, videos, but also flow fields [12]. Guo *et al.* [13], for instance, used a standard encoder-decoder CNN to predict the laminar steady state flow field around various types of geometries. Santos *et al.* [8] employed a Res-Unet-type architecture to model the flow in porous media. Similar approaches can be found in [14] and [15] for the prediction of flow fields around

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

airfoils. All of the aforementioned studies trained the respective models with numerical data, mostly from RANS simulations.

When it comes to wind turbine wakes, recently, Renganathan et al. [16] employed CNNs in combination with other machine learning methods to predict the two-dimensional mean velocity field downstream of a wind turbine based on low dimensional input data (meteorological measurements and SCADA data). Here, LiDAR measurements served as training data for the two-dimensional flow field prediction. Others used classical fully-connected neural networks to model velocity deficits and wake-added turbulence, while training the models with RANS simulation data [17, 18].

In this work we present a first feasibility study of a CNN-based surrogate model, hereafter referred to as WakeNet. The model is trained with the statistics obtained from actuator line LES runs that are efficiently computed using a GPU-based Lattice Boltzmann LES solver [19, 20]. As opposed to previous studies, the model predicts the full three-dimensional flow field downstream of the turbine and only relies on user-specified input data, comprising the mean inflow conditions and aerodynamic turbine data. The usability of the trained model is thus similar to common engineering wake models. A key feature is that it leverages a blade-element momentum approach (BEM) as a pre-processing step that provides the spatial distribution of the turbine's aerodynamic forces as an additional input feature.

The rest of the paper is organised as follows. In Section 2 we describe the general concept and architecture of the model. Section 3 outlines the LES set-up used for the simulation of the training data. In Section 4 we discuss the accuracy of the trained model and provide examples of the flow fields generated by the model. Final conclusions and an outlook of future work is given in Section 5.

2. WakeNet

The general objective of the model is to predict the time-averaged three-dimensional flow field downstream of a single turbine, while only relying on easily specifiable input data, i.e. the mean inflow conditions, aerodynamic data of the turbine (radial distributions of thickness, chord length, etc., as well as airfoil polars) and the tip-speed ratio (TSR). In the following we outline the overall modelling concept as well as a brief description of the CNN model architecture. An illustration of the workflow and model architecture is shown in Fig. 1. For a more comprehensive description of CNNs we refer to LeCun et al. [21, 10].

2.1. Input and Output Data

In in line with similar studies, e.g., [8, 13, 14], the model is set-up as a fully convolutional neural network. Hence, input data, all hidden feature maps, and output data are grid-like arrays that are solely processed via convolutional operations (no fully connected layer). In this case, the model input is two-dimensional and refers to the undisturbed mean inflow velocities ($\bar{u}, \bar{v}$ and $\bar{w}$ in the stream-wise, lateral and vertical direction x, y and z, respectively) and the turbulence intensity Ti on a cross-stream plane of $4\,D \times 4\,D$ (D being the turbine diameter) upstream of the turbine. Moreover, the model is provided with an estimate of the mean normal and tangential aerodynamic forces of the turbine ($\bar{p}_n$ and $\bar{p}_t$) that are mapped onto the same plane with an actuator-disk-like approach. To this end, we employ the BEM method fed with the inflow velocity $\bar{u}$, aerodynamic data of the turbine, and a user-specified TSR. On the input plane, $\bar{p}_n$ and $\bar{p}_t$ are thus only non-zero in the rotor-swept area of the turbine (see Fig. 1). Furthermore, it should be emphasized that the force data only serves as an input to the CNN that shall facilitate the prediction of the wake flow. It does not represent an output of the model. The accuracy of the forces, e.g., with respect to the corresponding forces in the simulated training data, is secondary. The mapped forces should merely be well-correlated with the characteristics of the

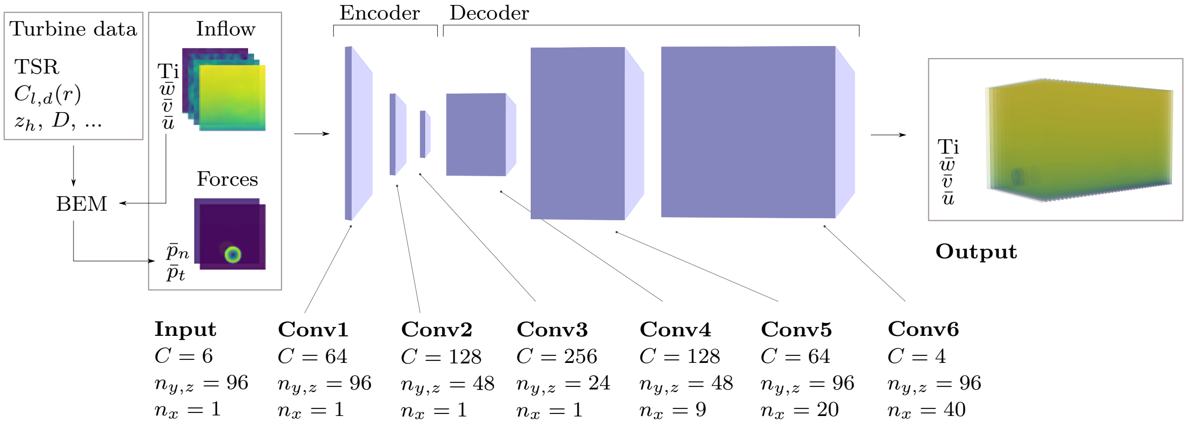

Figure 1: Illustration of the WakeNet model architecture and processing of the input data. The number of channels C and dimensions of the data refer to the output of the convolution in each layer.

wake flow. In summary, this results in a two-dimensional input with $n_y = n_z = 96$ points in the lateral and vertical direction, respectively, as well as $C_{\text{in}} = 6$ input channels.

The output of the model is the three-dimensional flow field downstream of the turbine, starting from the rotor plane. Similar to the input data, the flow field comprises the three mean velocity components $(\bar{u}, \bar{v}, \bar{w})$ and the turbulence intensity, Ti, $(C_{\text{out}} = 4)$ and has the same in-plane resolution n_y and n_z. The resolution in the stream-wise direction is $n_x = 40$. With a spacing of $\Delta x = 6\,\Delta y$ the output field spans $10\,D$ downstream. Note that one key idea of the model is the compatibility of the model output and input in terms of in-plane dimensions and parameters. At a future stage, this way, the output of one model instance can serve as the input to another instance, allowing for the modelling of entire farms (given a sufficient training of the model with wake inflow data). For a better conditioning of both input and output data, the velocity scales and forces are normalised with the mean hub height velocity u_h and $\rho\, u_h^2\, D$, respectively (with ρ being the density).

2.2. Model Architecture and Optimisation

The model architecture is a standard shared encoder-decoder structure as traditionally used for image-processing tasks with CNNs. The objective of the encoder is to sequentially increase the feature depth of the data while reducing the in-plane dimensions. To that end, each layer of the encoder comprises a convolution followed by a batch normalisation [22], ReLU activation, and a final down-sampling via MaxPooling. The kernel size of each convolution $k_y \times k_z$ is 3×3 with a stride $s_{y,z} = 1$. Subsequently, the decoder sequentially upsamples the encoded data using transpose convolutions. For this particular model, the upsampling not only increases the in-plane dimension (n_y and n_z), but also extends the data in the stream-wise direction x. After the respective transpose convolutions of the first two layers of the decoder (Conv4 and Conv5; see Fig. 1), we apply the same operations as the encoder (convolution, batch normalisation, and ReLU). In the last layer (Conv6) we employ just a convolution in order to transform the data to the desired output format. The kernel size of the convolutions of the decoder is $3 \times 3 \times 3$ with a stride $s_{x,y,z} = 1$. The total network comprises about $4.8 \cdot 10^6$ tunable parameters. Furthermore, it should be noted that larger networks with the same architecture (more channels and/or additional convolutional layers) did not yield better results for the cases considered in this work.

The model loss J is measured as the mean squared error (MSE) of $\bar{u}, \bar{v}, \bar{w}$ and Ti (L2 loss).

Table 1: Parameters covered in the simulations of the training dataset.

Parameter	n	
TSR	5	5.0, 6.0, 7.0, 7.5, 8.0
z_h/D	3	0.71, 0.95, 1.19
$u_h/\mathrm{m\,s^{-1}}$	5	5.0, 7.5, 10.0, 12.5, 15.0
z_0/m	3	10^{-3}, 10^{-2}, 10^{-1}
Ti	4	0.01, 0.05, 0.10, 0.15

The model is trained by minimising J via mini-batch gradient-descent using the Adam optimiser with decoupled weight decay (AdamW; [23]), with a learning rate $\gamma = 10^{-3}$ and a weight decay $\lambda = 10^{-2}$. The implementation of the model framework is based on the open-source deep learning library `pytorch` [24].

3. The Dataset

The comprehensive dataset for the training and testing of the model is obtained from LES of a single turbine in various inflow and operating conditions. The numerical set-up and an overview of the simulated cases is given below.

3.1. LES set-up

The simulations are performed with a GPU-based LBM (Lattice Boltzmann Method) solver [19]. As a collision model we use the cumulant LBM with parametrised relaxation rates, recovering the weakly-compressible Navier-Stokes equations with second-order accuracy in advection and fourth-order accuracy in diffusion [25]. The sub-grid scales are modelled with the anisotropic minimum dissipation model (AMD; [26]) with a model constant $C_i = 1/12$. The turbine is represented by an actuator line model (ALM). The ALM implementation closely follows the original method by Sørensen and Shen [27] and has previously been validated in [28, 20].

The domain measures $L_x = 14\,D$ and $L_y = L_z = 6\,D$ in the stream-wise, lateral and vertical direction, respectively. The resolution of the isotropic Cartesian grid is $\Delta x/D = 1/24$ referring to a total of $7.09 \cdot 10^6$ grid points. The simulations are run at a Mach number of 0.1. For the sake of simplicity, the inflow is prescribed at the inlet in terms of a mean logarithmic velocity profile that is perturbed with synthetic Mann turbulence [29]. The domain is periodic in y. The bottom boundary condition is a simple bounce-back scheme coupled to the iMEM wall modelling approach [30]. At the top, we use a symmetry boundary condition (simple bounce-forward, see [31]). An extrapolation boundary condition is applied at the outlet, combined with a viscous sponge layer. The turbine is located at $\boldsymbol{x} = \{3\,D, 3\,D, z_h\}$, while the hub height z_h varies between the cases (see Section 3.2). Each simulation is initially run-up for one domain flow-through time $T = L_x/u_h$. After the spin-up, the flow statistics serving as training data are gathered for $10\,T$ in the respective cross-stream planes downstream of the turbine. A corresponding plane sampled $2\,D$ upstream of the turbine serves as the inflow model input for each case during the training.

3.2. Cases

For the simulations of the dataset we consider the NREL 5 MW reference wind turbine [32], with $D = 126\,\mathrm{m}$ and a rated wind speed $u_0 = 11.5\,\mathrm{m\,s^{-1}}$. The cases cover ranges of five parameters which notably affect the wake properties, i.e., the TSR, the hub height z_h, the mean hub height velocity u_h, the shear (parametrized via the roughness length z_0 in the logarithmic inflow velocity profile) and the inflow turbulence intensity. The explicit values of all parameters are summarised in Table 1. Taking into account all possible combinations of the parameters,

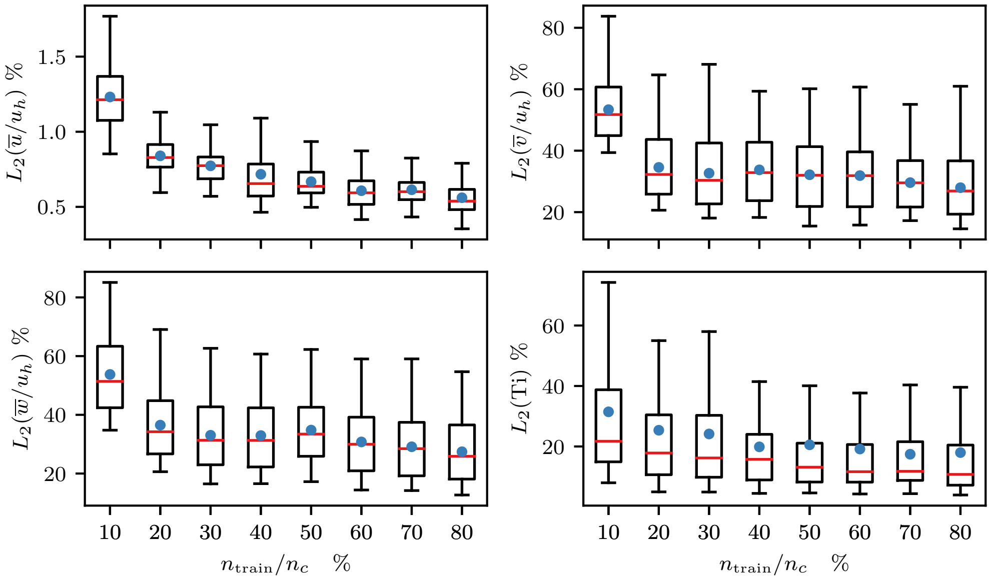

Figure 2: Box plots of the L2 relative test error of the model trained with different amounts of the training data n_{train}/n_c. The whiskers represent 1.5 times the interquartile range. The mean is given by blue dots (●).

the dataset contains a total amount of $n_c = 900$ cases. Each case ran in about 0.45 h on a single Nvidia RTX 2080 Ti GPU.

4. Results

The model is trained with the hyper-parameter settings given in Section 2. Due to the large size of the model and the individual training cases, the memory capacity of the utilised GPU (Nvidia RTX 2080 Ti) only accommodates a batch size of 4 for the training.

The overall accuracy of the model is evaluated in terms of the L2 relative error for each case i of the test data:

$$L_2 = \sqrt{\frac{\sum_{j=1}^{n_p} \left(Y_{\mathrm{WN},i}(\boldsymbol{x}_j) - Y_{\mathrm{LES},i}(\boldsymbol{x}_j)\right)^2}{\sum_{j=1}^{n_p} Y_{\mathrm{LES},i}(\boldsymbol{x}_j)^2}}\,, \tag{1}$$

where Y_{WN} and Y_{LES} are the WakeNet output and the LES reference value, respectively, and n_p is the number of grid points of the output data of each case, with $n_p = n_x\, n_y\, n_z$. Additional detailed local evaluations of the model will be discussed in Section 4.2.

4.1. Training

Initially, we investigate the sensitivity of the model to the number of training cases n_{train}. To this end, we perform several training runs with different amounts of training data. In every training, the model is evaluated with the same randomly chosen test data, referring to 20% ($n_{\mathrm{test}} = 180$) of the cases of the dataset. The model is trained for a maximum of 400 epochs, while the specific training length depends on the individual runs in order to avoid over-fitting. The training process with the largest amount of cases (80% of the dataset) took about 5.6 h on a desktop using the aforementioned GPU.

Fig. 2 provides a comparison of L_2 of the model trained different fractions of the dataset n_{train}/n_c. For all output parameters the model performance notably improves the larger the

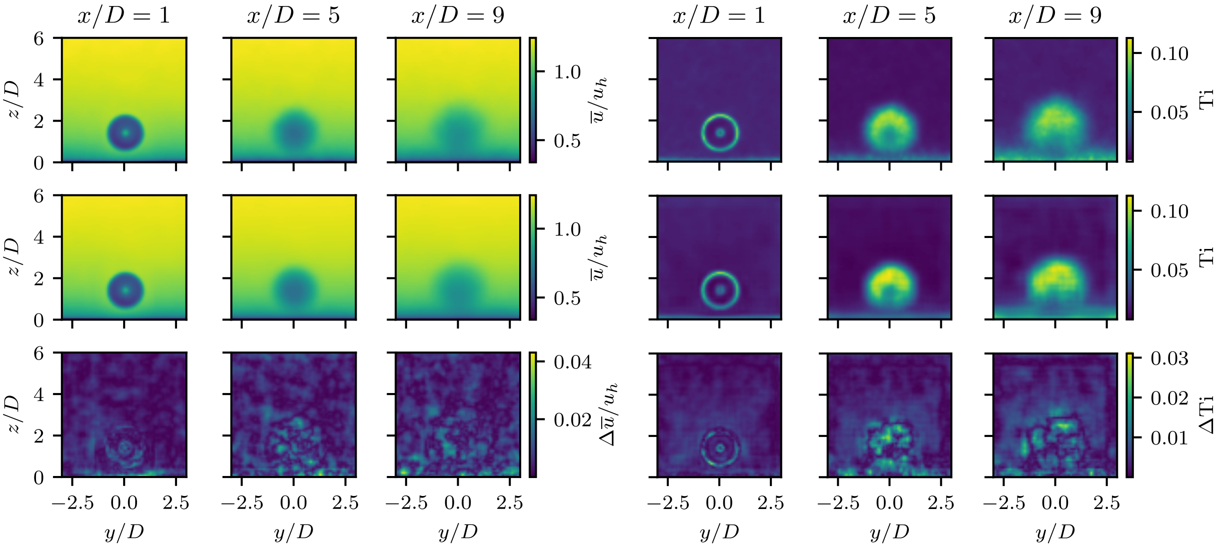

Figure 3: Contour plots of the LES data (top rows), model prediction (mid rows) and respective difference (bottom row) of $\bar{u}(y,z)/u_h$ (left) and $\mathrm{Ti}(y,z)$ (right) with TSR $= 7.0$, $z_h/D = 0.95$, $u_h = 7.5\,\mathrm{m\,s^{-1}}$, $z_0 = 0.1\,\mathrm{m}$ and $\mathrm{Ti} = 0.05$.

number of training cases. Only the error of Ti appears to converge already with 60% training data. For $\bar{u}/u_h$ and Ti, larger amounts of training data also reduce the spread of the test error. The model trained with 80% training data is able to reproduce the stream-wise velocity with a mean $L_2(\bar{u}/u_h)$ of 0.56%. The highest L_2 is found for the lateral and vertical velocity with mean errors of 27.95% and 27.41%, respectively, while the maximum errors lie around 60%. The turbulence intensity is predicted with a mean relative error of 17.01%. The observed accuracies show that the proposed model concept is generally able to parametrize the wake flow of the turbine. Particularly, the accuracy in reproducing the stream-wise velocity can arguably be appreciated. The larger relative errors in the other output variables seem to be mainly attributed to their lower order of magnitudes, as the absolute errors of $\bar{v}$, $\bar{w}$ and Ti are comparable to $\bar{u}$ or even lower (see below). Another factor can be the normalisation of $\bar{v}$, $\bar{w}$ and Ti using the mean stream-wise hub height velocity, u_h. Normalisations with other, more specific velocity scales might yield a better conditioning of the data and improve the training.

4.2. Flow Field Predictions

A more detailed analysis of the predictions of specific flow features is outlined in the following. For the sake of brevity we limit the discussion to the model trained with 80% of the dataset. As a starting point, we compare a selection of contour plots of the LES data against the model output for an exemplary case of the test data; see Fig. 3. The stream-wise velocity is well-captured at all downstream locations. This refers to the characteristic shape and strength of the velocity deficit as well as the velocity of the ambient flow. The largest differences measure about $0.04\,u_h$, while occurring in small patches scattered across the entire flow field. The model also reasonably predicts the overall characteristics of Ti throughout the entire wake. When compared to $\bar{u}$, larger deviations from the LES solution are more clearly clustered in the far-wake. Fig. 4 depicts the corresponding vertical profiles of all output variables of the same case as well as two additional cases with lower and higher TSR, respectively. Regardless of the TSR, the model excellently predicts the downstream evolution of $\bar{u}$. Thus, dependencies of the wake characteristics (e.g., magnitude of the velocity deficit, wake rotation, and the downstream recovery rate) on the thrust and power of the turbine are successfully encoded in the model.

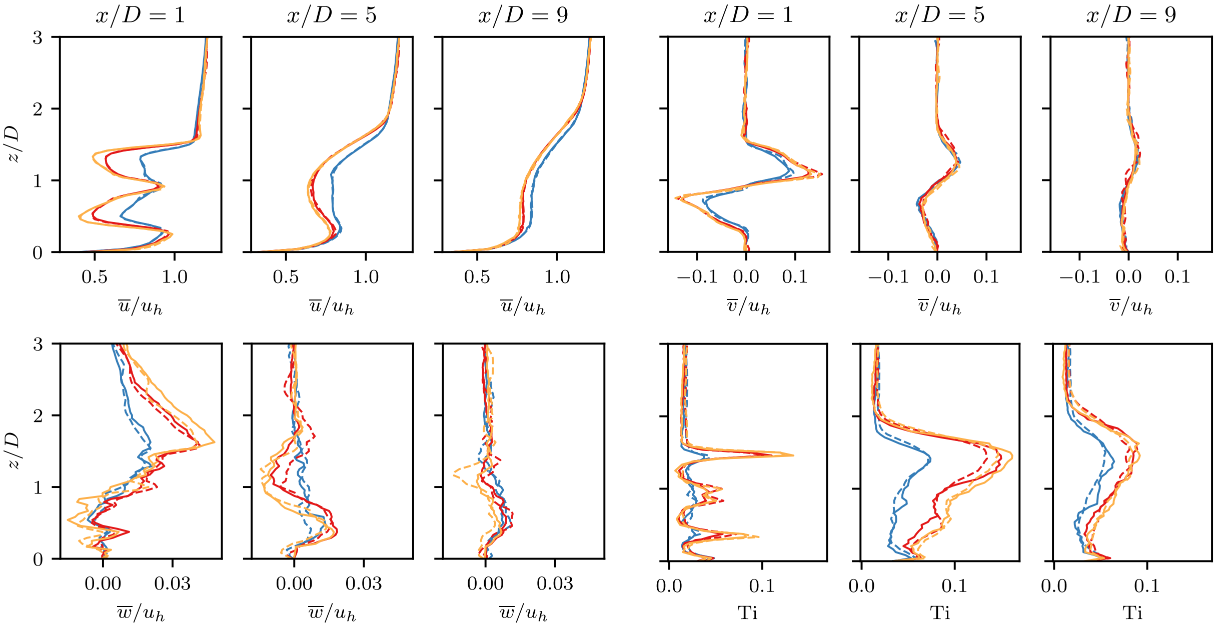

Figure 4: Vertical profiles of $\bar{u}, \bar{v}$, $\bar{w}$ and Ti as predicted by WakeNet (full lines) compared against the corresponding LES results (dashed lines). All cases refer to the same inflow conditions as shown in Fig. 3. The TSR is 5.0 (—/- -), 7.0 (—/- -) and 8.0 (—/- -), respectively.

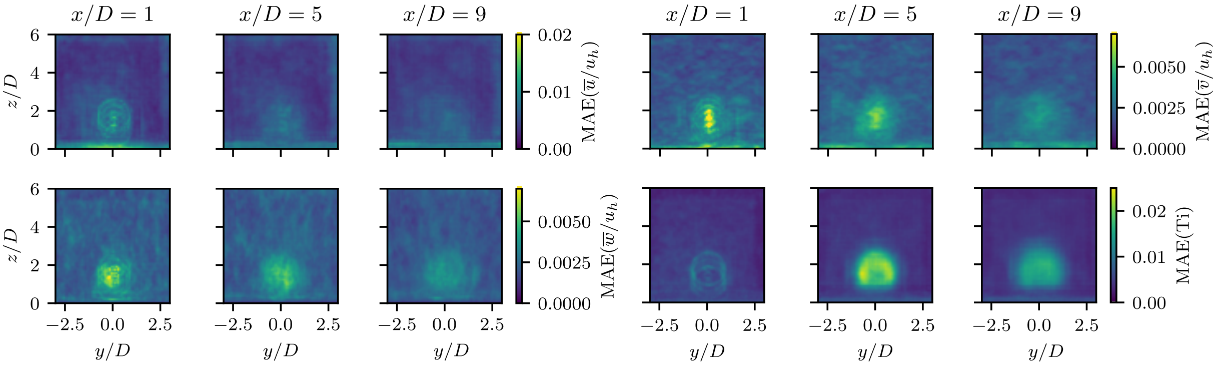

Figure 5: Contour plots of the local MAE of the test data.

This also applies to the characteristic change of the mean velocity profile from near to far-wake due to the laminar-turbulent transition. In absolute terms, a similarly good agreement is found for the lateral and vertical velocity. For all TSRs shown, the turbulence intensity is slightly overestimated in the far-wake. Still, the key features of Ti, including a distinct increase after the laminar-turbulent transition as well as the subsequent decay throughout the far-wake, are well predicted.

In order to analyse the spatial variations of the model accuracy, we compare the local mean absolute error

$$\mathrm{MAE}(\boldsymbol{x}) = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} |Y_{\mathrm{WN},i}(\boldsymbol{x}) - Y_{\mathrm{LES},i}(\boldsymbol{x})| \tag{2}$$

of the test data, in Fig. 5. The highest MAE of $\bar{u}$ is found in the near-wake ($x/D = 1$) and clearly outlines the outer edge of the rotor swept-area of the turbine. With increasing downstream distance the error tends to decrease. As for $\bar{v}$ and $\bar{w}$, the largest errors are also found in the near-wake, yet more concentrated in the center of the rotor-swept area. In contrast, the MAE of the turbulence intensity is largest at $x/D = 5$. Generally, we find that larger error magnitudes tend to occur in areas with a large variability of the corresponding variable in the training data. For instance, Ti is relatively small in the near-wake, regardless of the inflow conditions or TSR of the turbine, and typically increases drastically after the transition of the wake (see, e.g., Fig. 4). The downstream distance of this point of transition depends heavily on the inflow conditions and thrust of the turbine. Both magnitude and local distribution of Ti therefore vary mostly at $3 \leq x/D \leq 8$ between the training cases, while being relatively similar closer to the turbine.

5. Conclusion

We have presented a CNN-based model framework for the prediction of the three-dimensional steady-state wake flow downstream of a single turbine. This feasibility study shows that CNNs are a promising model family to encode high-fidelity simulation data into computationally efficient surrogate models. A single prediction of the trained model runs in 1326 µs on a single core of a local CPU (Intel i7-7800X) or in 33 µs on an Nvidia RTX 2080 Ti GPU, respectively. The discussed results indicate that the model is able to accurately capture detailed features of the wake flow under different inflow and operating conditions. This combination of high efficiency and accuracy makes it a potential contender for established low to mid fidelity approaches such as analytical wake models or even RANS.

While the presented results are arguably promising, it should also be stressed that the discussed model refers to the most simple CNN architecture possible, namely a shared encoder-decoder. Judging from other applications of CNNs to fluid flow problems, further improvements can be anticipated using, e.g., an encoder with separated decoder for each output variable [15], ResNet [33] or ResUNet-type architectures [8] and/or locally weighted loss functions, to name a few. Another aspect worth investigating is the sensitivity of the model to wake conditions not covered in the training data. Or, in other words, how well does a model trained on a certain dataset generalise to other flow scenarios? This not only includes other inflow or operating conditions but also different turbine types.

Code and Data Availability

The source code of the model is available for download at `https://source.coderefinery.org/wind_energy_uu/wakenet`. The utilised training data can be found at `https://doi.org/10.5281/zenodo.5911015`.

References

[1] Veers P, Dykes K, Lantz E, Barth S, Bottasso C L, Carlson O, Clifton A, Green J, Green P, Holttinen H, Laird D, Lehtomäki V, Lundquist J K, Manwell J, Marquis M, Meneveau C, Moriarty P, Munduate X, Muskulus M, Naughton J, Pao L, Paquette J, Peinke J, Robertson A, Sanz Rodrigo J, Sempreviva A M, Smith J C, Tuohy A and Wiser R 2019 *Science* **366** eaau2027

[2] Porté-Agel F, Bastankhah M and Shamsoddin S 2020 *Boundary-Layer Meteorology* **174** 1–59

[3] Jiang C, Vinuesa R, Chen R, Mi J, Laima S and Li H 2021 *Physics of Fluids* **33** 055133

[4] Mohan A T, Tretiak D, Chertkov M and Livescu D 2020 *Journal of Turbulence* **21** 484–524

[5] Yang X I A, Zafar S, Wang J X and Xiao H 2019 *Physical Review Fluids* **4** 034602

[6] Balasubramanian A G, Guastoni L, Güemes A, Ianiro A, Discetti S, Schlatter P, Azizpour H and Vinuesa R 2021 *arXiv:2107.07340* (*Preprint* 2107.07340)

[7] Han R, Wang Y, Zhang Y and Chen G 2019 *Physics of Fluids* **31** 127101

[8] Santos J E, Xu D, Jo H, Landry C J, Prodanović M and Pyrcz M J 2020 *Advances in Water Resources* **138** 103539

[9] Jin X, Cai S, Li H and Karniadakis G E 2021 *Journal of Computational Physics* **426** 109951

[10] LeCun Y, Bengio Y and Hinton G 2015 *Nature* **521** 436–444

[11] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (MIT Press)

[12] Vinuesa R and Brunton S L 2021 *arXiv:2110.02085 [physics]* (*Preprint* `2110.02085`)

[13] Guo X, Li W and Iorio F 2016 Convolutional Neural Networks for Steady Flow Approximation *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA) pp 481–490

[14] Sekar V, Jiang Q , Shu C and Khoo B C 2019 *Physics of Fluids* **31** 057103

[15] Bhatnagar S, Afshar Y, Pan S, Duraisamy K and Kaushik S 2019 *Computational Mechanics* **64** 525–545

[16] Renganathan S A, Maulik R, Letizia S and Iungo G V 2021 *arXiv:2109.02411 [physics]* (*Preprint* `2109.02411`)

[17] Ti Z, Deng X W and Yang H 2020 *Applied Energy* **257** 114025

[18] Purohit S, Ng E Y K and Syed Ahmed Kabir I F 2022 *Renewable Energy* **184** 405–420

[19] Janßen C F, Mierke D, Überrück M, Gralher S and Rung T 2015 *Computation* **3** 354

[20] Asmuth H, Olivares-Espinosa H and Ivanell S 2020 *Wind Energy Science* **5** 623–645

[21] Lecun Y, Bottou L, Bengio Y and Haffner P 1998 *Proceedings of the IEEE* **86** 2278–2324

[22] Ioffe S and Szegedy C 2015 *arXiv:1502.03167 [cs]* (*Preprint* `1502.03167`)

[23] Loshchilov I and Hutter F 2019 *arXiv:1711.05101 [cs, math]* (*Preprint* `1711.05101`)

[24] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J and Chintala S 2019 PyTorch: An Imperative Style, High-Performance Deep Learning Library *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc.) pp 8024–8035

[25] Geier M, Pasquali A and Schönherr M 2017 *J. Comput. Phys.* **348** 862–888

[26] Rozema W, Bae H J, Moin P and Verstappen R 2015 *Physics of Fluids* **27** 085107

[27] Sørensen J N and Shen W Z 2002 *J. Fluids Eng.* **124** 393–399

[28] Asmuth H, Olivares-Espinosa H, Nilsson K and Ivanell S 2019 *Journal of Physics: Conference Series* **1256** 012022

[29] Mann J 1998 *Probabilist. Eng. Mech.* **13** 269–282

[30] Asmuth H, Janßen C F, Olivares-Espinosa H and Ivanell S 2021 *Physics of Fluids* **33** 105111

[31] Krüger T, Kusumaatmaja H, Kuzmin A, Shardt O, Silva G and Viggen E M 2016 *The Lattice Boltzmann Method - Principles and Practice* (Heidelberg, Germany: Springer)

[32] Jonkman J, Butterfield S, Musial W and Scott G 2009 Definition of a 5-MW Reference Wind Turbine for Offshore System Development Tech. Rep. NREL/TP-500-38060 NREL

[33] Chung T, Da Wang Y, Armstrong R T and Mostaghimi P 2020 *Transport in Porous Media* **135** 25–37