Machine-Learning-Based Surrogate Modeling of Aerodynamic Flow Around Distributed Structures

Jincheng Zhang[*] and Xiaowei Zhao[†]
University of Warwick, Coventry, England CV47AL, United Kingdom

https://doi.org/10.2514/1.J059877

A machine-learning-based surrogate modeling method for distributed fluid systems is proposed in this paper, where a dimensionality reduction technique is used to reduce the flowfield dimension and a regression model is used to predict the reduced coefficients from the input parameters. The surrogate modeling method is specifically designed to tackle the fluid systems involving distributed aerodynamic structures, and its performance is illustrated by the application on the wake flow around wind turbine arrays in an atmospheric boundary layer. The main idea is to first decompose the whole fluid domain into subdomains, then carry out surrogate modeling for each subdomain by treating both the boundary information and the distributed flow parameters as the input parameters, and finally obtain the whole flowfield by combining the flowfield of each subdomain with the consideration of the matching condition at the subdomain interface. The proposed surrogate modeling method is applied to two test cases: a one-dimensional Poisson equation and a high-fidelity wind farm wake model. The results demonstrate the great efficiency and accuracy of the surrogate model and its excellent scalability to distributed systems of different scales.

Nomenclature

$\boldsymbol{a}$	=	bias term in autoencoder
$\boldsymbol{b}$	=	bias term in neural network regressor
D	=	rotor diameter
d	=	distributed parameters
$\boldsymbol{g}$	=	forward transform of dimensionality reduction
$\hat{\boldsymbol{g}}^{-1}$	=	inverse transform of dimensionality reduction
I	=	flow quantities at subdomain interface
K	=	number of distributed structures
l	=	one-dimensional domain length
$\mathcal{M}$	=	regression model
N	=	size of training samples
N_h	=	hidden-layer neuron number
N_r	=	dimension of the reduced coefficients
q	=	heat source term
S	=	magnitude of the heat source
T	=	temperature
T_L	=	temperature at the left boundary
T_R	=	temperature at the right boundary
$\mathcal{U}$	=	flowfield by computational fluid dynamics simulations
$\mathcal{U}^r$	=	reduced representations of the flowfield
$\hat{\mathcal{U}}$	=	flowfield predicted by surrogate model
$\boldsymbol{v}$	=	weight matrix in autoencoder
$\boldsymbol{w}$	=	weight matrix in neural network regressor
$\boldsymbol{Z}$	=	collection of flowfields
α	=	proper orthogonal decomposition coefficient
$\boldsymbol{\alpha}$	=	vector of proper orthogonal decomposition coefficients
α_{NN}	=	L2 regularization coefficient in neural network training
β	=	thermal diffusivity
γ	=	wind turbine yaw angle
ϵ_{all}	=	overall prediction error
ϵ_{mr}	=	dimensionality reduction error
μ	=	input parameter
$\boldsymbol{\nu}$	=	proper orthogonal decomposition basis
σ	=	neural network activation function
Ω	=	parameter space
$\mathbb{I}$	=	indicator function

I. Introduction

COMPUTATIONAL fluid dynamics (CFD) is an important tool for investigating complex flow problems. It has been used in many engineering and scientific applications, e.g., aircraft design, weather forecasting, and turbulence research. However, despite the fast development of high-performance computing (HPC) technology, the use of CFD for repetitive and real-time tasks (such as optimization and real-time control) is still highly challenging due to the requirement of enormous computational resources and long simulation time. For instance, the three-dimensional (3-D) Reynold-averaged Navier–Stokes simulation typically requires hundreds/thousands of CPU hours, while large eddy simulations (LESs) and the direct numerical simulations require even more computational resources. Therefore, surrogate modeling, which aims at constructing an efficient yet accurate approximation to the full CFD model, has attracted a lot of attention, such as in aeroelastic computations [1], aerodynamic load evaluations [2], uncertainty quantification of turbulence models [3], and combustion modeling [4].

Reduced basis methods, such as proper orthogonal decomposition (POD) [5–7] and dynamic mode decomposition [8], have been used in surrogate modeling. The basic idea of surrogate modeling using reduced basis methods is to decompose the flow dynamics into several reduced bases, which can be interpreted as the coherent structures or modes of the flow. Then, the original high-dimensional flowfield can be represented by the reduced coefficients of a much lower dimension, with each dimension representing a mode. This type of method has advantages in physical interpretations and can be used to analyze the energy content and the frequency of the coherent structures. On the other hand, due to the fast development and democratization of machine learning algorithms, surrogate modeling techniques based on machine learning are getting attention in recent years, such as in the aerodynamic simulations with consideration of multiple operating conditions [9] and in the simulations of single-injector combustion process [10].

The surrogate modeling using machine learning techniques can be classified into two categories. One approach is to directly formulate the surrogate modeling of fluid systems as a supervised machine learning problem and employ the state-of-the-art machine learning technique (e.g., deep learning) to train the model [11,12]; then, the trained machine learning model can serve directly as the emulator of the original CFD model. This approach can be quite effective if enough training data are available. However, the generation of training data is very expensive because a lot of high-fidelity simulations are required to

Received 10 June 2020; revision received 12 September 2020; accepted for publication 18 October 2020; published online 28 December 2021.

[*] Marie Curie Early Stage Researcher, School of Engineering; jincheng.zhang@warwick.ac.uk.

[†] Professor, School of Engineering; xiaowei.zhao@warwick.ac.uk (Corresponding Author).

cover the input parameter space. Another approach is to first reduce the high-dimension flowfield into its low-dimension representation; then, a supervised machine learning problem is formulated to predict the reduced coefficients of the flow dynamics given the input parameters. The latter approach is the focus of our paper because it usually requires less training data than the former one.

The combination of POD and supervised machine learning has been investigated recently. In Ref. [13], Hesthaven and Ubbiali proposed a surrogate modeling technique called POD-NN, where POD was used to extract the reduced basis and a neural network (NN) was used for approximating the map between flow parameters and the POD coefficients. The combination of POD and other machine learning techniques (i.e., Gaussian process regression) was investigated in Refs. [14,15], and it was tested on a set of numerical examples. The potential to incorporate physical constraints was investigated in Ref. [16]. They proposed the use of particular solutions in the POD expansion to enforce certain constraints, e.g., boundary conditions. In addition, a set of supervised machine learning techniques was considered, including NN, multivariate polynomial regression, k nearest neighbors, and decision trees. All the aforementioned work employed POD for dimensionality reduction.

The general framework of the surrogate modeling method developed in this paper is shown in Fig. 1, where $\{\mu_0, \mu_1, \ldots, \mu_N\}$, $\{\mathcal{U}_0, \mathcal{U}_1, \ldots, \mathcal{U}_N\}$, and $\{\mathcal{U}_0^r, \mathcal{U}_1^r, \ldots, \mathcal{U}_N^r\}$ represent the training samples of the input parameters, the corresponding CFD flowfields, and the reduced coefficients of the flowfields; and μ_{test}, $\mathcal{U}_{\text{test}}^r$, and $\hat{U}_{\text{test}}$ represent the input parameter of interest, the predicted reduced coefficients of the flow, and the full flowfield prediction. Various dimensionality reduction techniques can be used in this framework as long as both the forward and the inverse transforms are available. The surrogate modeling framework consists of the offline and online stages. In the offline stage, as shown by the solid, arrowed lines, multiple CFD simulations are carried out to generate the flowfield data using a set of input parameters obtained by a sampling strategy (e.g., Latin hypercube sampling). The generated flowfields are then reduced to their low-dimension representations by a dimensionality reduction technique, and a regression model is constructed to approximate the mapping from the input parameters to the reduced coefficients. The regression model can be a simple curve fitting or a complex supervised machine learning model. As for the dimensionality reduction, it can be achieved by either traditional reduced basis methods or machine learning techniques. In fact, dimensionality reduction in machine learning shares a lot of similarities with model reduction in scientific computing. For example, principal component analysis in machine learning is equivalent to POD in scientific computing in certain senses [7]. Once the offline stage is completed, the online stage, as shown by the dashed, arrowed lines, is carried out by propagating the input parameter of interest to the reduced coefficients through the regression model and then predicting the flowfield by the inverse transform of the dimensionality reduction process. In this paper, three-dimensionality reduction techniques arising from both scientific computing and machine learning are investigated, including POD, independent component analysis (ICA) [17], and autoencoder (AE) [18].

Another novelty of this paper is that it extends the proposed surrogate modeling method to tackle distributed flow problems. Distributed fluid systems are quite common in daily life and industrial applications, such as the natural convection of a heater array in heat exchangers [19,20], the distributed roughness elements in boundary-layer control [21], the heat transfer of a building array in a turbulent boundary layer [22], and the wake interactions of wind turbines within a wind farm [23–25]. The numerical simulation of such systems usually requires a lot of computational resources, and the optimization/control of such systems is very difficult because the repetitive evaluations of CFD models with distributed flow parameters are needed. This motivates the work in this paper on the surrogate modeling of distributed systems, which has not been investigated yet in the literature. A suitable surrogate modeling method for distributed systems should have the following features:

1) The surrogate model can simulate the fluid system with different flow parameters and preferably different layouts.

2) The method should be scalable such that the surrogate model can simulate the distributed system of different scales.

In this paper, a scalable surrogate modeling method for distributed fluid systems is proposed, where the whole fluid domain is first decomposed into subdomains; then, the surrogate modeling for each subdomain is carried out by treating both the boundary information and the distributed flow parameters as the input parameters; finally, the whole flowfield is obtained by coupling the flowfield of each subdomain altogether with the consideration of the matching condition at the subdomain interface. The information exchange at the subdomain interface needs to be tackled specifically according to the types of the problems. Here, an iterative updating process is introduced for diffusion-dominant problems since each subdomain has an impact on all the other subdomains, whereas a sequential prediction process is sufficient for convection-dominant problems because the impact of the downstream structures on the upstream flow can be ignored.

Two test cases are used to demonstrate the efficiency, accuracy, and scalability of the proposed surrogate modeling method. The first one is the one-dimensional (1-D) Poisson equation, which is used to represent the application of the proposed method to diffusion problems. Then, a large-scale industrial application (the surrogate modeling of wind farms) is investigated. The simulator for onshore/offshore wind farm applications (SOWFA) [26], an LES solver developed for the 3-D flow simulation around a wind turbine array, is used to generate the high-fidelity data. The application in wind farms aims at capturing the wake interactions between wind turbines, which have a large impact on the plant's overall performance. In the literature, a range of models has been developed to investigate the wake interactions [27], including the high-fidelity LES models and low-fidelity analytical models. In high-fidelity simulations, the turbine rotors were usually represented as actuator disks [28] or actuator lines [29]. The simulations were carried out in Refs. [30,31] using both methods. Although the high-fidelity simulations can predict the turbine wakes accurately, they are too slow for wind farm control design. Low-fidelity wake models are still the main tools for fast wake predictions. Such models include the Jensen model [32,33], the Frandsen model [34], the FLOw Redirection and Induction in Steady State (FLORIS) model [35], and the model proposed in Ref. [36]. To retain the reliability of the high-fidelity model while achieving much faster predictions, data-driven surrogate modeling provides an alternative way to model wind farms. The surrogate modeling of a single turbine was studied in Refs. [37,38], where the wake prediction was presented. In Ref. [39], the surrogate modeling of two turbine cases was investigated, but the scalability to the wind farm was not considered. In this paper, we apply the proposed surrogate modeling approach for the challenging issue of wind farm wake modeling. The results show that the surrogate model predicts the wind farm wake flow efficiently and accurately, with the root-mean-squared error being only 2% of the freestream wind speed.

The remaining part of this paper is organized as follows: the machine-learning-based surrogate modeling method for distributed fluid systems is described in Sec. II. The proposed method is applied to a diffusion-dominant problem (more specifically, 1-D Poisson equation) and a convection-dominant problem (more specifically, wind farm simulations) in Sec. III to demonstrate its scalability, efficiency, and accuracy. Finally, the conclusions are drawn in Sec. IV.

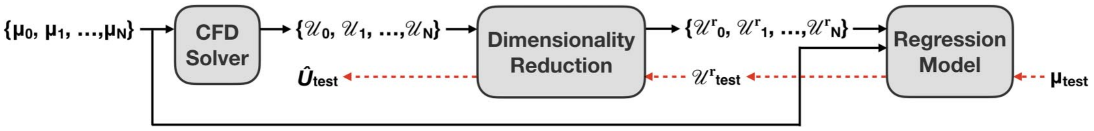


Fig. 1 The general framework of the machine-learning-based surrogate modeling.

II. Machine-Learning-Based Surrogate Modeling of Distributed Fluid Systems

In machine learning, dimensionality reduction is usually used to reduce high-dimension training input into its low-dimension representation before feeding it into a regression/classification model. For surrogate modeling of fluid systems, however, the goal is to predict the high-dimension flowfield giving a few input parameters. The direct construction of a regression model to predict the high-dimension flowfield is prone to overfitting, especially when there are not enough training data available. Therefore, the idea of surrogate modeling for fluid systems is to first reduce the high-dimension flowfield into their low-dimension representation, then predict the reduced coefficients from the input parameters, and finally reconstruct the flowfield based on the reduced coefficients. One way to achieve this is the use of reduced basis methods (e.g., POD) as is done in Refs. [13–16].

We briefly describe the reduced basis method and then illustrate the equivalence of reduced basis method and other machine-learning-based dimensionality reduction techniques in the context of surrogate modeling. Given N samples of input parameters $[\mu_0, \mu_1, \dots, \mu_N]$ and the corresponding CFD flowfields $\boldsymbol{Z} = [\mathcal{U}_0, \mathcal{U}_1, \dots, \mathcal{U}_N]$, the POD basis $\{\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \dots, \boldsymbol{\nu}_k, \dots\}$ can be constructed by the singular value decomposition

$$\boldsymbol{Z} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{W}^T \tag{1}$$

where $\boldsymbol{\nu}_k$ is the kth column vector of $\boldsymbol{V}$. Then, after choosing the number of the POD basis as N_r, the flowfield can be approximated by

$$\tilde{U}(\mu_i) = \sum_{k=1}^{N_r} \alpha_k(\mu_i)\boldsymbol{\nu}_k \tag{2}$$

where the POD coefficients $\boldsymbol{\alpha}(\mu_i) = [\alpha_1(\mu_i), \alpha_2(\mu_i), \dots, \alpha_{N_r}(\mu_i)]$ are the reduced coefficients of the original flowfield $\mathcal{U}_i$. Then, a regression model $\mathcal{M}$ can be trained based on the training input $[\mu_0, \mu_1, \dots, \mu_N]$ and the training target $[\boldsymbol{\alpha}(\mu_0), \boldsymbol{\alpha}(\mu_1), \dots, \boldsymbol{\alpha}(\mu_N)]$ such that

$$\mathcal{M}(\mu_i) \approx \boldsymbol{\alpha}(\mu_i) \tag{3}$$

After training, the prediction of the flowfield can then be given by

$$\hat{U}(\mu_{\text{test}}) = \sum_{k=1}^{N_r} [\mathcal{M}(\mu_{\text{test}})]_k \boldsymbol{\nu}_k \tag{4}$$

where μ_{test} is the input parameter of interest. In fact, the reduced basis method here can be viewed as a dimensionality reduction technique in machine learning, where the forward transform $\boldsymbol{g}$ and the inverse transform $\hat{\boldsymbol{g}}^{-1}$ are defined as

$$[\boldsymbol{g}(\mathcal{U})]_k = \langle \mathcal{U}, \boldsymbol{\nu}_k \rangle, \quad 1 \le k \le N_r \tag{5}$$

$$\hat{\boldsymbol{g}}^{-1}(\boldsymbol{\alpha}) = \sum_{k=1}^{N_r} \alpha_k \boldsymbol{\nu}_k \tag{6}$$

where $\langle\,\rangle$ denotes the inner product. From the preceding formulation, it is clear that the use of reduced basis methods in surrogate modeling can be replaced by any other dimensionality reduction techniques in machine learning as long as there exist both the forward transform $\boldsymbol{g}$, which maps the flowfield into the reduced coefficients, and the inverse transform $\hat{\boldsymbol{g}}^{-1}$, which maps the reduced coefficients to the approximation of the flowfield. After dimensionality reduction and regression model training, the flowfield can then be predicted by

$$\hat{U}(\mu_{\text{test}}) = \hat{\boldsymbol{g}}^{-1}(\mathcal{M}(\mu_{\text{test}})) \tag{7}$$

The proposed surrogate modeling procedure is summarized as Algorithm 1. In this work, the fully connected NN with one hidden layer is chosen as the regression model. POD, ICA, and AE are employed for dimensionality reduction. Hereby, these methods are referred as POD-NN, ICA-NN, and AE-NN. The machine learning packages Scikit-learn [40] and Keras [41] are used to facilitate the implementation of the proposed algorithm.

Algorithm 1: The surrogate modeling method using machine learning

```
% Offline process
Generate $N$ samples of the input parameters: $[\mu_0, \mu_1, \dots, \mu_N]$.
Run the high-fidelity CFD solver $N$ times to generate the flowfield
    $\boldsymbol{Z} = [\mathcal{U}_0, \mathcal{U}_1, \dots, \mathcal{U}_N]$.
Use a chosen dimensionality reduction technique to obtain the reduced
    coefficients $[\mathcal{U}^r_0, \mathcal{U}^r_1, \dots, \mathcal{U}^r_N]$ and the corresponding forward and inverse
    transforms $\boldsymbol{g}$ and $\hat{\boldsymbol{g}}^{-1}$.
Train the regression model $\mathcal{M}$ using the training input $[\mu_0, \mu_1, \dots, \mu_N]$
    and the training target $[\mathcal{U}^r_0, \mathcal{U}^r_1, \dots, \mathcal{U}^r_N]$.
% Online process
Set the input parameter of interest $\mu_{\text{test}}$.
Predict the flowfield as $\hat{\boldsymbol{g}}^{-1}(\mathcal{M}(\mu_{\text{test}}))$.
```

A. Regression Model

The regression model used in this work is a fully connected NN with one hidden layer, as illustrated in Fig. 2; and the corresponding input–output relation can be expressed as

$$\begin{aligned} \boldsymbol{h} &= \sigma(\boldsymbol{w}_1\boldsymbol{x} + \boldsymbol{b}_1), \\ \boldsymbol{y} &= \boldsymbol{w}_2\boldsymbol{h} + \boldsymbol{b}_2 \end{aligned} \tag{8}$$

where $\boldsymbol{w}_1$, $\boldsymbol{b}_1$, $\boldsymbol{w}_2$, and $\boldsymbol{b}_2$ are the training variables; and σ is the activation function. N_1, N_2, and N_3 in Fig. 2 represent the input dimension, the hidden-layer neuron number, and the output dimension. The NN training process involves the updating of the corresponding weights w_{ij} based on the gradient $\partial J/\partial w_{ij}$, where J is the objective function to be minimized. Automatic differentiation is employed to calculate the gradients. The mean-squared error (MSE) between the target and the NN output is chosen as the objective function where a L2 regularization term is further added in order to tackle overfitting. The Adam optimization algorithm [42] is used for NN training in this work.

B. Dimensionality Reduction

Three types of dimensionality reduction techniques (i.e., POD, ICA, and AE) are employed in this paper to demonstrate the feasibility of using any dimensionality reduction technique with an inverse transform in the proposed surrogate modeling framework. As most of the previous studies were based on POD, this work investigates the use of ICA and AE as the alternative dimensionality reduction technique in the context of surrogate modeling. ICA is a popular technique first developed for signal processing, and it can be used for general purpose dimensionality reduction. The interested readers may refer to Ref. [43] for more details. AE is a machine learning

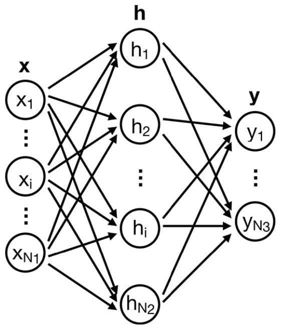


Fig. 2 The illustration of a fully connected NN with one hidden layer.

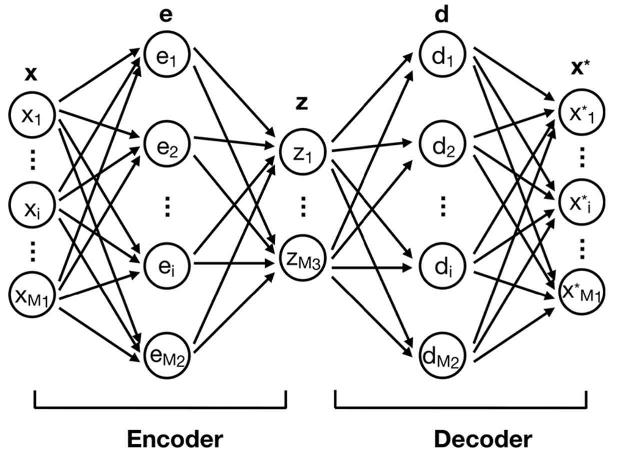


Fig. 3 The illustration of an AE with three hidden layers in the NN structure.

technique that makes use of NN to encode a high-dimension input to a low-dimension latent space and then decode it back. It consists of an encoder NN and a decoder NN, and it is trained in a self-supervised manner. The AE used in this work is illustrated in Fig. 3, where three hidden layers are included in the NN structure. The encoder part can be expressed as

$$
\begin{aligned}
\boldsymbol{e} &= \sigma(\boldsymbol{v}_1\boldsymbol{x} + \boldsymbol{a}_1),\\
\boldsymbol{z} &= \boldsymbol{v}_2\boldsymbol{e} + \boldsymbol{a}_2
\end{aligned}
\tag{9}
$$

where $\boldsymbol{v}_1$, $\boldsymbol{a}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{a}_2$ are the training variables of the encoder; whereas the decoder part can be expressed as

$$
\begin{aligned}
\boldsymbol{d} &= \sigma(\boldsymbol{v}_3\boldsymbol{z} + \boldsymbol{a}_3),\\
\boldsymbol{x}^* &= \boldsymbol{v}_4\boldsymbol{d} + \boldsymbol{a}_4
\end{aligned}
\tag{10}
$$

where $\boldsymbol{v}_3$, $\boldsymbol{a}_3$, $\boldsymbol{v}_4$, and $\boldsymbol{a}_4$ are the training variables of the decoder. M_1, M_2, and M_3 in Fig. 3 represent the original data dimension, the hidden-layer neuron number, and the reduced data dimension. In this work, for simplicity, the hidden-layer neuron numbers of the encoder and the decoder are assumed to be the same and are set as twice of the reduced dimension. The NN training is carried out using the Adam optimization algorithm to minimize the MSE between the NN output and its target value. The target value is the same as the input of the training dataset; thus, it is termed as self-supervised training.

C. Extension to Distributed Flow Problems

The surrogate modeling method is extended to tackle distributed flow problems here. A typical example of a distributed fluid system is illustrated in Fig. 4, where M cylinders of diameters $\{d_1, d_2, \ldots, d_M\}$ are positioned in a rectangular flow domain. The cylinder diameter is the distributed parameter in this example and, in fact, the distributed parameter can be an array of any other properties of the structures (the surface roughness, thermal conductivity, etc.). The surrogate modeling of this problem is formulated as how to construct a model to predict the whole flowfield around the cylinder array given the values of $d_1, d_2, \ldots, d_M$. The method proposed in the previous section can be used directly by treating $[d_1, d_2, \ldots, d_M]$ as the input parameter μ. However, this approach has two fundamental flaws:

1) The training of the surrogate model requires a significant number of CFD evaluations because the whole flow domain simulation is regarded as a single training sample and the dimension of μ is very high, such that a large sample size is required in order to cover the input parameter space.

2) The so-constructed surrogate model is not scalable because it can only be used to simulate the distributed system of the same scale as the training samples.

The scalable surrogate modeling method proposed in our paper can solve these issues. First, the whole flow domain is decomposed into subdomains, with each subdomain containing one distributed structure, as illustrated by the dashed rectangle in Fig. 4. Then, the surrogate modeling is carried out for each subdomain by treating both the distributed parameter of the structure inside the subdomain and the boundary information as the input parameter μ. Finally, the flowfields of all the subdomains are combined together with the consideration of the matching condition at the subdomain interface. The proposed approach is inspired naturally by the domain decomposition in high-performance computing in CFD, where the whole domain/mesh is divided into subdomains/blocks and each Message Passing Interface (MPI) thread handles only its assigned subdomain with the interface information exchange between MPI threads. The approach can be viewed as employing the surrogate model of individual subdomain to replace the task of each MPI thread, thus greatly reducing the online prediction time. The whole surrogate modeling procedure is summarized as Algorithm 2, where an iterative process is introduced to update the flow quantities at the interface in order to enforce the physical constraints at the interface. The detailed updating rule in line 12 of Algorithm 2 is problem dependent. Two numerical examples are given in the rest of the paper with the detailed implementation of the updating rules.

III. Numerical Results

The application of the proposed method on the 1-D Poisson equation and on wind farm wake modeling is described in this section. The first test case demonstrates the accuracy, efficiency, and scalability of the proposed surrogate modeling method in diffusion-dominant problems. The second test case further demonstrates the ability of the proposed method in modeling large-scale fluid systems.

A. Application on 1-D Poisson Equation

1. Problem Setup

The 1-D steady-state heat transfer with distributed heat sources under consideration is illustrated in Fig. 5, where a 1-D domain of length l is shown with K uniformly distributed heat sources of magnitudes $[S_1, S_2, \ldots, S_K]$. It can be described by the following equation:

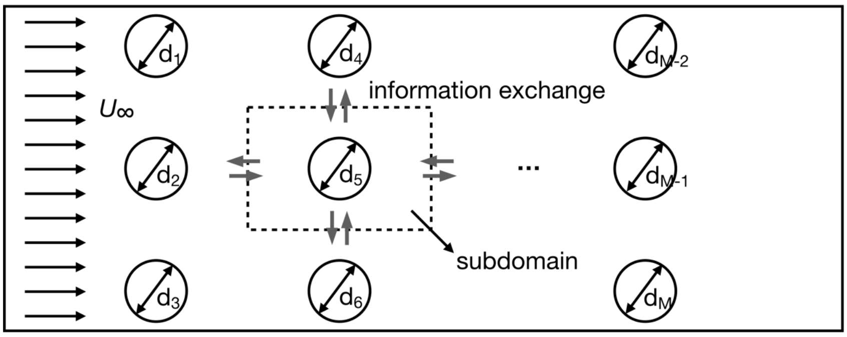


Fig. 4 A typical example of a distributed fluid system.

Algorithm 2: The surrogate modeling method for distributed flow problems

```
% **Offline process**
Generate $N$ samples of the input parameters: $[\mu_0, \mu_1, \ldots, \mu_N]$, where $\mu_i = [d_i, I_i]$. Note that $d_i$ represents the distributed parameters of a single structure and $I_i$ represents the flow quantities at the subdomain boundary.
Run the high-fidelity CFD solver multiple times to obtain the flowfields in a single subdomain $[\mathcal{U}_0, \mathcal{U}_1, \ldots, \mathcal{U}_N]$, where $\mathcal{U}_i$ is generated with the distributed parameter $d_i$ and the boundary condition is given by $I_i$.
Carry out surrogate modeling using Algorithm 1. Then, the flowfield in a single subdomain $\hat{U}_{\text{test}}$ can be predicted by the surrogate model given $[d_{\text{test}}, I_{\text{test}}]$.
% **Online process**
Set the distributed parameters of all the $K$ structures: $[\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_K]$.
Initialize the flow quantities at the boundary of each subdomain $[\tilde{I}_1, \tilde{I}_2, \ldots, \tilde{I}_K]$.
**while** True, **do**
  **for** $j = 1$ to $K$, **do**
    Predict the flowfield $\tilde{U}_j$ in $j$th subdomain given the input parameter $\tilde{d}_j$ and boundary information $\tilde{I}_j$.
  **end for**
  Update $[\tilde{I}_1, \tilde{I}_2, \ldots, \tilde{I}_K]$ based on the surrogate model prediction from both sides of the interfaces.
  **if** the changes of $[\tilde{I}_1, \tilde{I}_2, \ldots, \tilde{I}_K]$ are very subtle, **then**
    The updating process converges. **Break**.
  **end if**
**end while**
Combine $[\tilde{U}_1, \tilde{U}_2, \ldots, \tilde{U}_K]$ to obtain the prediction of the whole flowfield.
```

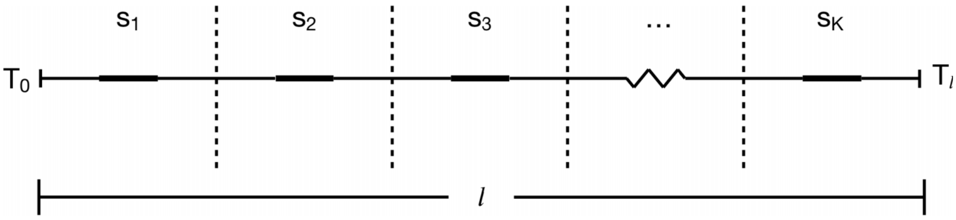


Fig. 5 The 1-D steady-state heat transfer problem under consideration.

$$-\beta \frac{\partial^2 T}{\partial x^2} = q(x) \tag{11}$$

where the heat source term

$$q = \sum_{i=1}^{K} S_i \mathbb{I}_{[i-(2/3)](l/K)<x<[i-(1/3)](l/K)}$$

and the boundary condition is given as $T(0) = T_0$ and $T(l) = T_l$. Here, $\mathbb{I}$ represents the indicator function. The surrogate modeling of this problem aims at predicting the temperature field efficiently, given the boundary conditions (T_0 and T_l) and the distributed parameters $[S_1, S_2, \ldots, S_K]$.

In the following, the length of each subdomain l/K is set as one and the thermal diffusivity β is set as one. The FTCS (which stands for forward-time central space) scheme is implemented for numerically solving the equation. The mesh-dependence study shows that a uniform mesh of size $1/80$ is sufficient. All calculations are deemed convergent when the root-mean-square error (RMSE) of the temperature profiles between two consecutive time steps is less than 10^{-6}.

2. *Results*

First, the surrogate modeling of a subdomain with a single heat source is carried out. Four hundred samples of the input parameters $[\mu_0, \mu_1, \ldots, \mu_{400}]$ in the parameter space

$$\Omega = [-5.0, 5.0] \times [-5.0, 5.0] \times [-5.0, 5.0]$$

are generated using Latin hypercube sampling, where $\mu_i = [S_i, T_{Li}, T_{Ri}]$ with S_i, T_{Li}, and T_{Ri} representing the magnitude of the heat source, the temperature at the left boundary, and the temperature at the right boundary of the ith sample. Then, the Poisson equation in a domain of length l/K is solved numerically for each sample of the input parameters to generate the temperature profiles. The generated data are then split into two parts: 320 training and validation samples and 80 test samples.

Three surrogate modeling methods (i.e., POD-NN, ICA-NN, and AE-NN) are employed to build surrogate models to predict the temperature profile with the magnitude of the heat source, the temperature at the left boundary, and the temperature at the right boundary as the model input. The temperatures at both boundaries and heat sources are scaled to zero mean and unit variance separately before feeding into the NN for training. The learning rate is set as 10^{-3}. A grid-search procedure is carried out to determine the optimal hyperparameters in the dimensionality reduction and regression models, based on four-fold cross-validation errors. The optimal hyperparameters, the dimensionality reduction errors, and the prediction errors are given in Table 1 for all three methods, where N_r represents the dimension of the reduced coefficient, N_h represents the hidden-layer neuron number of the NN regressor, and α_{NN} represents the L2 regularization coefficient of the NN regressor. The optimal hyperparameter is chosen from the parameter space

$$\Omega_{N_r \times N_h \times \text{ActFun} \times \alpha_{\text{NN}}} = \{2, 3, 4, 5, 6, 7\} \times \{4, 6, 8, 10, 12\} \times \{\text{tanh}, \text{relu}\} \times \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$$

The dimensionality reduction error ϵ_{mr} is defined as the RMSE between $\hat{\boldsymbol{g}}^{-1}(\boldsymbol{g}(\mathcal{U}_{\text{test}}))$ and $\mathcal{U}_{\text{test}}$; and the prediction error ϵ_{all} is

Table 1 The optimal hyperparameters of the three surrogate modeling methods for the 1-D Poisson case

Surrogate modeling method	N_r	N_h	Activation function	α_{NN}	ϵ_{mr}	ϵ_{all}
POD-NN	3	10	relu	10^{-6}	2.47×10^{-5}	2.68×10^{-3}
ICA-NN	3	12	relu	10^{-1}	2.47×10^{-5}	5.86×10^{-2}
AE-NN	3	8	relu	10^{-2}	2.41×10^{-3}	9.30×10^{-3}

relu = rectified linear unit.

defined as the RMSE between $\hat{g}^{-1}(\mathcal{M}(\mu_{\text{test}}))$ and $\mathcal{U}_{\text{test}}$, where μ_{test} is the set of test parameters and $\mathcal{U}_{\text{test}}$ is the corresponding temperature profiles obtained by the numerical solver. The comparisons of the temperature profiles between the surrogate model predictions and numerical solutions for four randomly selected test cases are shown in Fig. 6, including the results given by POD-NN, ICA-NN, and AE-NN. As can be seen, the temperature profiles given by POD-NN match perfectly with the ones given by the numerical solver. Therefore, POD-NN is used in the following for the surrogate modeling of the distributed systems.

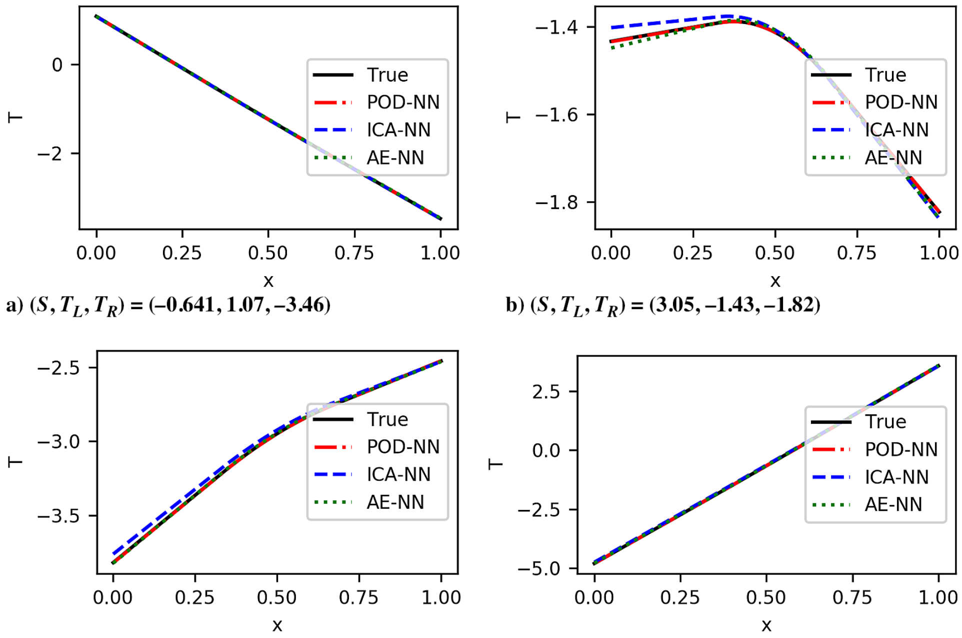
For the prediction of the temperature profile of a whole domain with distributed heat sources, the initialization and updating of the temperature at the subdomain boundary need to be specified. Here, the temperatures are initialized randomly between T_0 and T_l for the internal subdomain interfaces; and the boundary conditions T_0 and T_l are imposed for the most left and the most right subdomains. Then, the updating process aims at matching T and $\partial T/\partial x$ at the subdomain interface, because it can be derived that the temperature and the first-order derivative of the temperature need to be continuous at the interface.

The results for the case of the domain length of $l = 5$ with $K = 5$ heat sources are given here to illustrate the surrogate model's scalability. One hundred samples of the distributed parameters and the corresponding numerical solutions are generated in order to assess the accuracy of the surrogate model. The comparisons of the temperature profiles between the surrogate model predictions and numerical solutions for four randomly selected test cases are shown in Fig. 7. The RMSE of the surrogate model predictions compared to the high-fidelity numerical solutions averaged over all the test samples is 3.77×10^{-2}. The results clearly demonstrate that the proposed surrogate model predicts the distributed heat transfer problem efficiently and accurately.

B. Surrogate Modeling of Wind Farms

1. Wind Farm Model

The high-fidelity flowfield data are needed for the surrogate modeling of wind farms. In this work, the simulator for onshore/offshore wind farm applications is employed for CFD data generation. The SOWFA is a numerical solver developed based on OpenFOAM for the 3-D large-eddy simulation of wind flow around wind



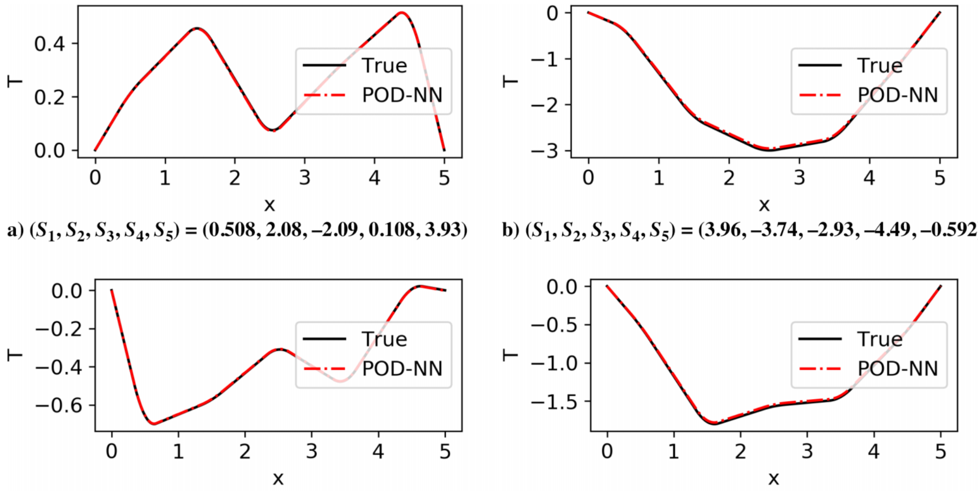
a) (S, T_L, T_R) = (−0.641, 1.07, −3.46)

b) (S, T_L, T_R) = (3.05, −1.43, −1.82)

c) (S, T_L, T_R) = (2.66, −3.82, −2.46)

d) (S, T_L, T_R) = (−0.697, −4.81, 3.58)

Fig. 6 The prediction results of a subdomain with a single heat source.



a) $(S_1, S_2, S_3, S_4, S_5)$ = (0.508, 2.08, −2.09, 0.108, 3.93)

b) $(S_1, S_2, S_3, S_4, S_5)$ = (3.96, −3.74, −2.93, −4.49, −0.592)

c) $(S_1, S_2, S_3, S_4, S_5)$ = (−4.70, −0.432, 1.49, −2.22, 1.76)

d) $(S_1, S_2, S_3, S_4, S_5)$ = (0.909, −4.76, 0.589, −2.41, −0.849)

Fig. 7 The prediction results of a domain with five uniformly distributed heat sources.

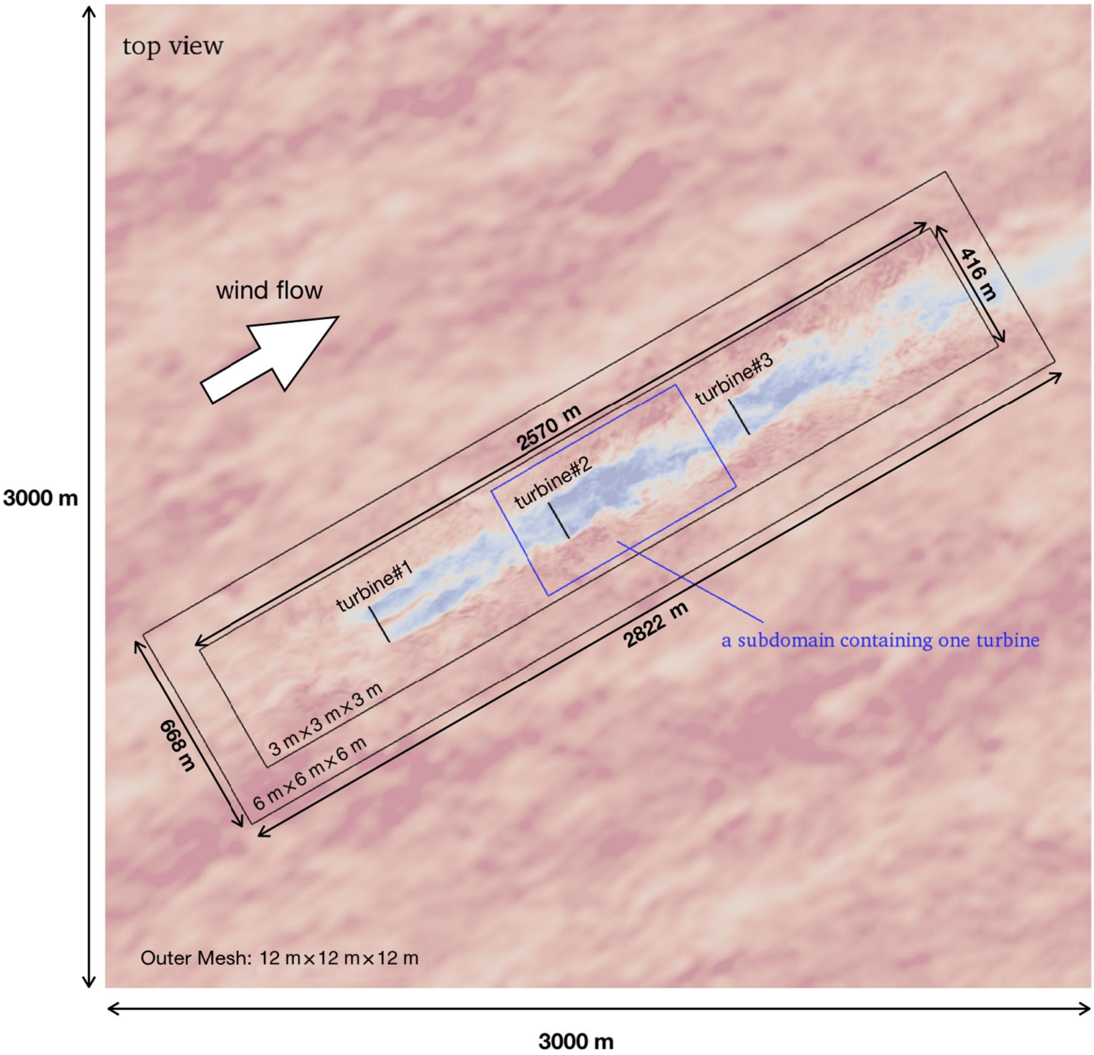


Fig. 8 A top view of the simulation domain at turbine hub height.

turbine array in the atmospheric boundary layer, where the turbine rotors are represented by actuator disks or actuator lines. The detailed implementations and the validations of the SOWFA can be found in Refs. [29,44].

For wind farm simulations, a precursor simulation of a neutral atmospheric boundary layer is first carried out to obtain the initial flowfield and inflow boundary conditions; then, wind farms are simulated using actuator line method (ALM). A top view of the simulation domain at hub height is given in Fig. 8. The size of the simulation domain is $3000 \times 3000 \ \times 3000$ m, with the inflow wind coming from a southwest direction. For the mesh generation, a two-level local mesh refinement is used, as is suggested in Ref. [45]. The outer mesh dimension is $12 \times 12 \times 12$ m, the inner mesh dimension is $3 \times 3 \times 3$ m, and the dimension of the mesh in between is $6 \times 6 \times 6$ m. The total number of cells is 1.8×10^7. In this way, the mesh size around the turbine rotors is 3 m so that the simulation can capture the detailed turbine wake dynamics. Three National Renewable Energy Laboratory 5 MW baseline turbines are positioned in the simulation domain with a five-rotor-diameter spacing in the downstream direction. The rotor diameter of this baseline turbine (denoted as D hereafter) is 126.4 m. For each simulation case, 1500 s simulations are carried out with a time step of 0.02 s. Each case requires around 44 h using 256 processors in the local HPC clusters. After LESs, the mean velocity field is obtained by averaging the instantaneous flowfield from 400 to 1400 s. The surrogate modeling in the following part aims at predicting the two-dimensional (2-D) mean velocity field at turbine hub height efficiently, given the inflow conditions and the yaw angles of all the turbines $[\gamma_1, \gamma_2, \ldots, \gamma_K]$ for a wind farm consisting of K turbines.

2. *Results*

First, the surrogate modeling of a subdomain containing a single turbine is carried out. For the high-fidelity data generation, the wind farm simulation is carried out for the case with three turbines in a row, as is illustrated in Fig. 8. Three inflow wind velocities (i.e., 8, 9, and 10 m/s) are considered with a freestream turbulence intensity of 6%. For each wind velocity, 30 samples of turbine yaw angles $[\gamma_0, \gamma_1, \ldots, \gamma_{30}]$ in the parameter space

$$\Omega = [-30.0°, 30.0°] \times [-30.0°, 30.0°] \times [-30.0°, 30.0°]$$

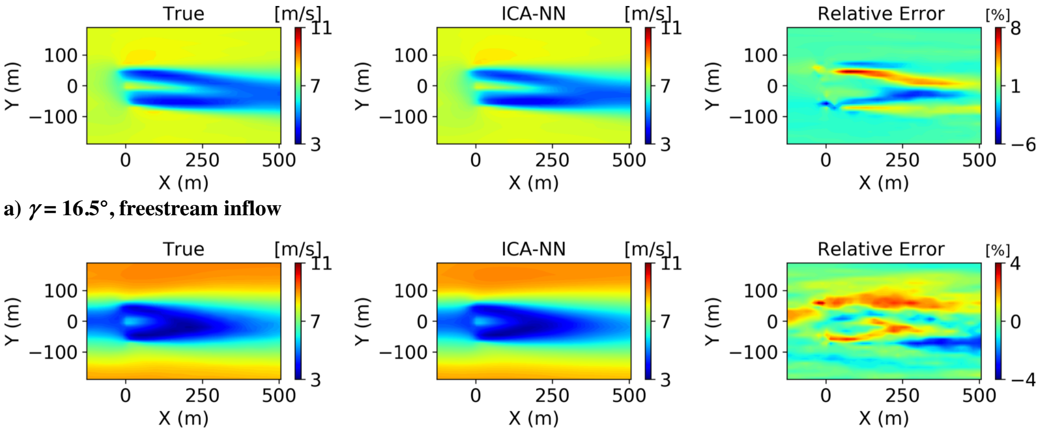
are generated using Latin hypercube sampling, where $\gamma_i = [\gamma_i^1, \gamma_i^2, \gamma_i^3]$ with γ_i^1, γ_i^2, and γ_i^3 representing the yaw angles of the first, second, and third turbines of the ith sample. Then, the SOWFA is used for generating the flowfield for each yaw setting. In total, 90 simulations are carried out. The flowfield of each subdomain containing one turbine is then extracted as a single data sample. One such subdomain is shown in Fig. 8. Therefore, 270 data samples are available for the surrogate modeling of the flowfield around one turbine. The data samples are then split into two parts: 216 training and validation samples, and 54 test samples.

After completing offline data generation, three surrogate modeling methods (i.e., POD-NN, ICA-NN, and AE-NN) are employed to build surrogate models to predict the flowfield in a single subdomain with the turbine yaw angle and the wind speeds at 30 uniformly distributed discrete points along the inflow boundary as the model input. The yaw angles and the inflow velocity profiles are scaled to zero mean and unit variance separately before feeding into the NN for training. The learning rate is set as 10^{-3}. A grid-search procedure is carried out to determine the optimal hyperparameters in the dimensionality reduction and regression models, based on four-fold cross-validation errors. The optimal hyperparameters, the dimensionality reduction errors, and the prediction errors are given in Table 2 for all three methods, where the optimal hyperparameter is chosen from the parameter space

$$\Omega_{N_r\times N_h\times \text{ActFun}\times\alpha_{\text{NN}}} = \{5, 10, 15, 20, 25, 30\}\times\{10, 20, 30, 40, 50\}\times\{\tanh, \text{relu}\}\times\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$$

Table 2 The optimal hyperparameters of the three surrogate modeling methods for the wind farm case

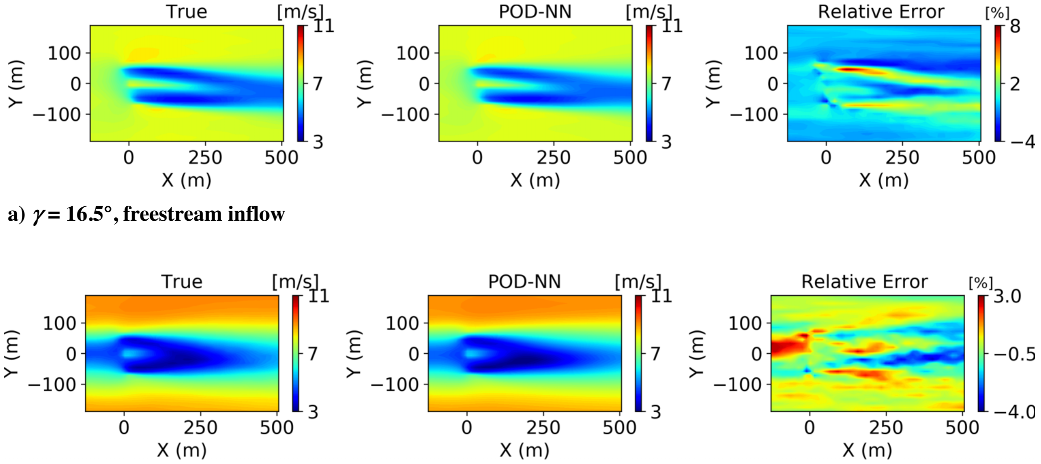
Surrogate modeling method	N_r	N_h	Activation function	α_{NN}	ϵ_{mr}	ϵ_{all}
POD-NN	15	50	relu	10^{-5}	5.35×10^{-2}	1.67×10^{-1}
ICA-NN	20	50	tanh	10^{-1}	4.45×10^{-2}	1.29×10^{-1}
AE-NN	15	50	relu	10^{-4}	7.74×10^{-2}	1.64×10^{-1}



a) $\gamma = 16.5°$, freestream inflow

b) $\gamma = 11.6°$, upstream wake

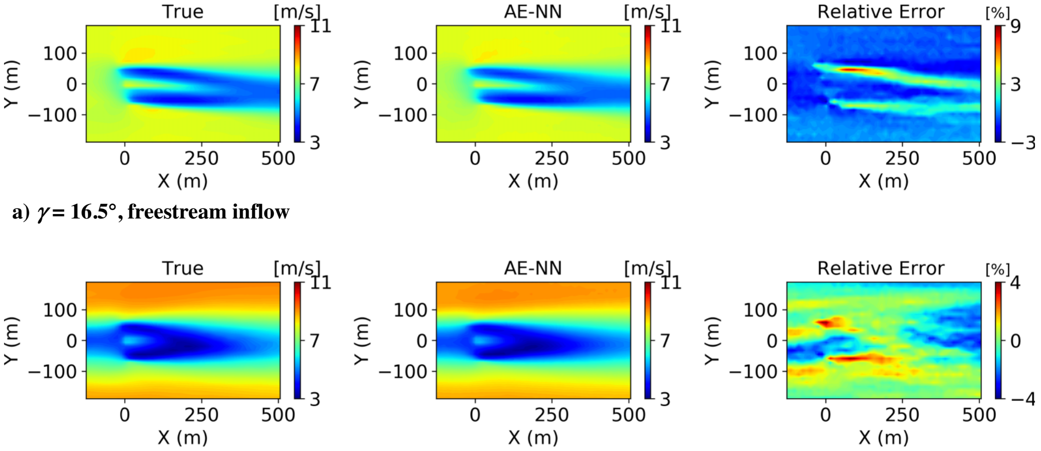
Fig. 9 The comparisons between ICA-NN predictions and high-fidelity simulations.



a) $\gamma = 16.5°$, freestream inflow

b) $\gamma = 11.6°$, upstream wake

Fig. 10 The comparisons between POD-NN predictions and high-fidelity simulations.



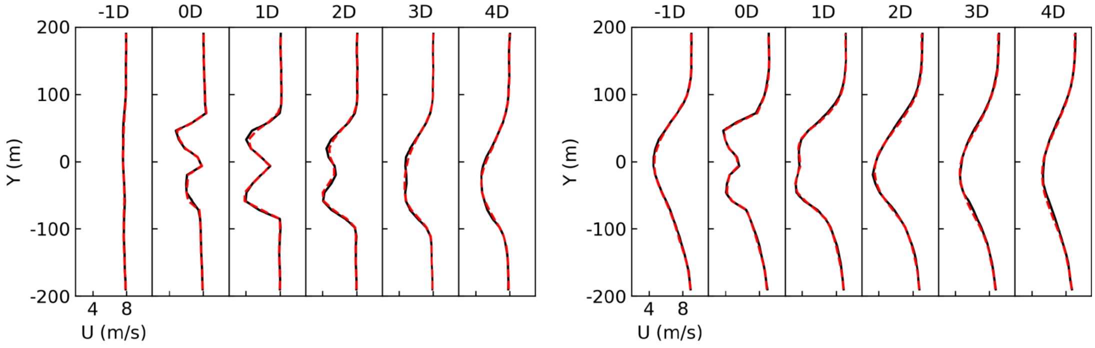
a) $\gamma = 16.5°$, freestream inflow

b) $\gamma = 11.6°$, upstream wake

Fig. 11 The comparisons between AE-NN predictions and high-fidelity simulations.

The dimensionality reduction error ϵ_{mr} is defined as the RMSE between $\hat{\boldsymbol{g}}^{-1}(\boldsymbol{g}(\mathcal{U}_{\text{test}}))$ and $\mathcal{U}_{\text{test}}$; and the prediction error ϵ_{all} is defined as the RMSE between $\hat{\boldsymbol{g}}^{-1}(\mathcal{M}(\mu_{\text{test}}))$ and $\mathcal{U}_{\text{test}}$, where μ_{test} is the set of test inflow conditions and yaw angles and $\mathcal{U}_{\text{test}}$ is the corresponding 2-D velocity fields obtained by the SOWFA.

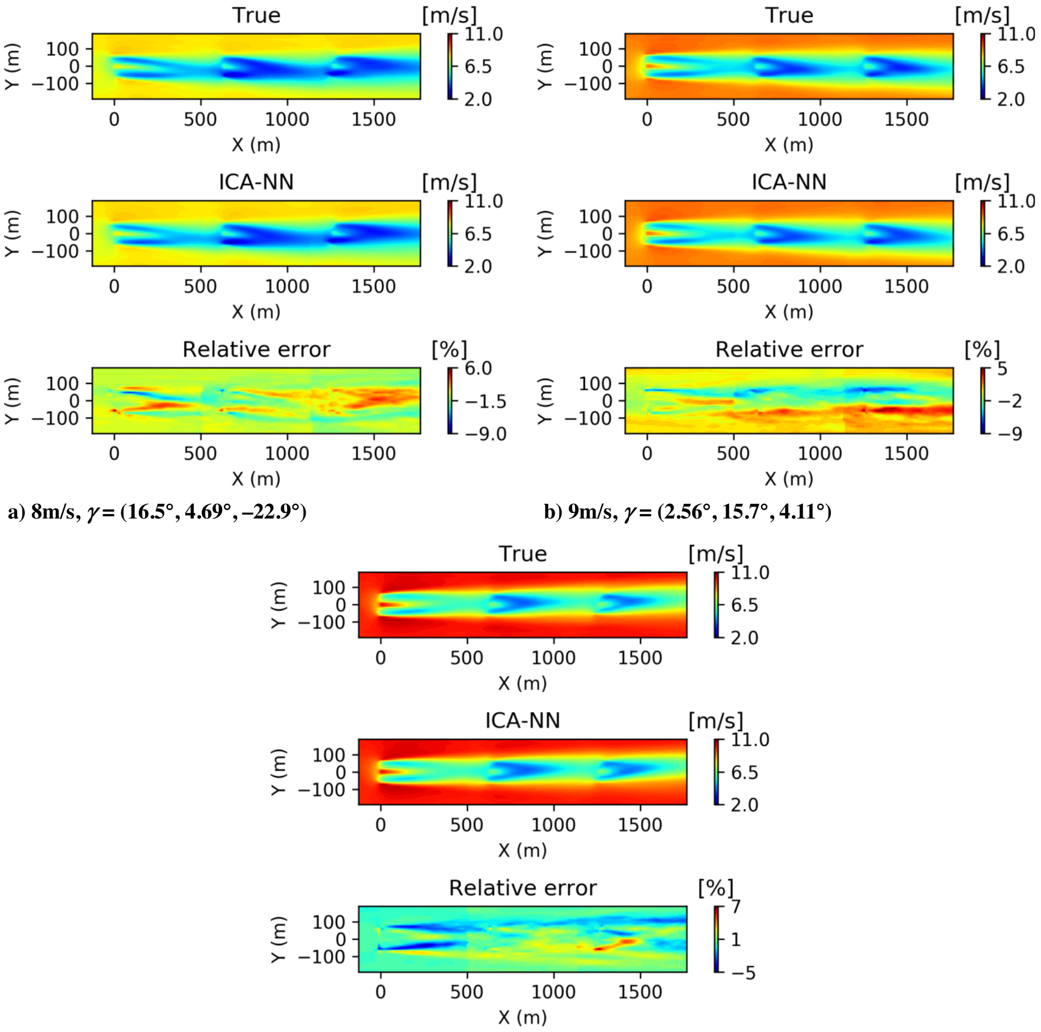
After NN training, the predictions of single turbine wakes are carried out and the results are compared with the SOWFA results for all 54 test cases. Two typical test cases (the first one with the turbine operating in freestream inflow and the second one with the turbine operating in the upstream wake) are shown in Figs. 9–11, where the flowfields are predicted by ICA-NN, POD-NN, and AE-NN, respectively. The relative error is defined as the absolute error of the surrogate model prediction divided by the inflow mean velocity. As can be seen, the overall flowfields predicted by all three methods match with the SOWFA results quite well. As the turbine wake evolves from upstream to downstream, the wake profiles at different streamwise locations (from one rotor diameter ($X = -1D$) in front of the turbine to four rotor diameters ($X = 4D$) behind the turbine) are investigated in order to further examine the prediction performance. As shown in Fig. 12a, a freestream inflow at $X = -1D$ travels downstream and hits the turbine rotor at $X = 0D$. The generated wake then travels from $X = 1D$ to $X = 4D$ in the deflected direction caused by the yaw effects while slowly recovering toward the freestream conditions and reaching a Gaussian-shape profile at $X = 4D$.



a) $\gamma = 16.5°$, freestream inflow

b) $\gamma = 11.6°$, upstream wake

Fig. 12 The velocity profiles for the one-turbine cases by ICA-NN (dashed) and high-fidelity simulations (solid).



a) 8m/s, $\gamma = (16.5°, 4.69°, -22.9°)$

b) 9m/s, $\gamma = (2.56°, 15.7°, 4.11°)$

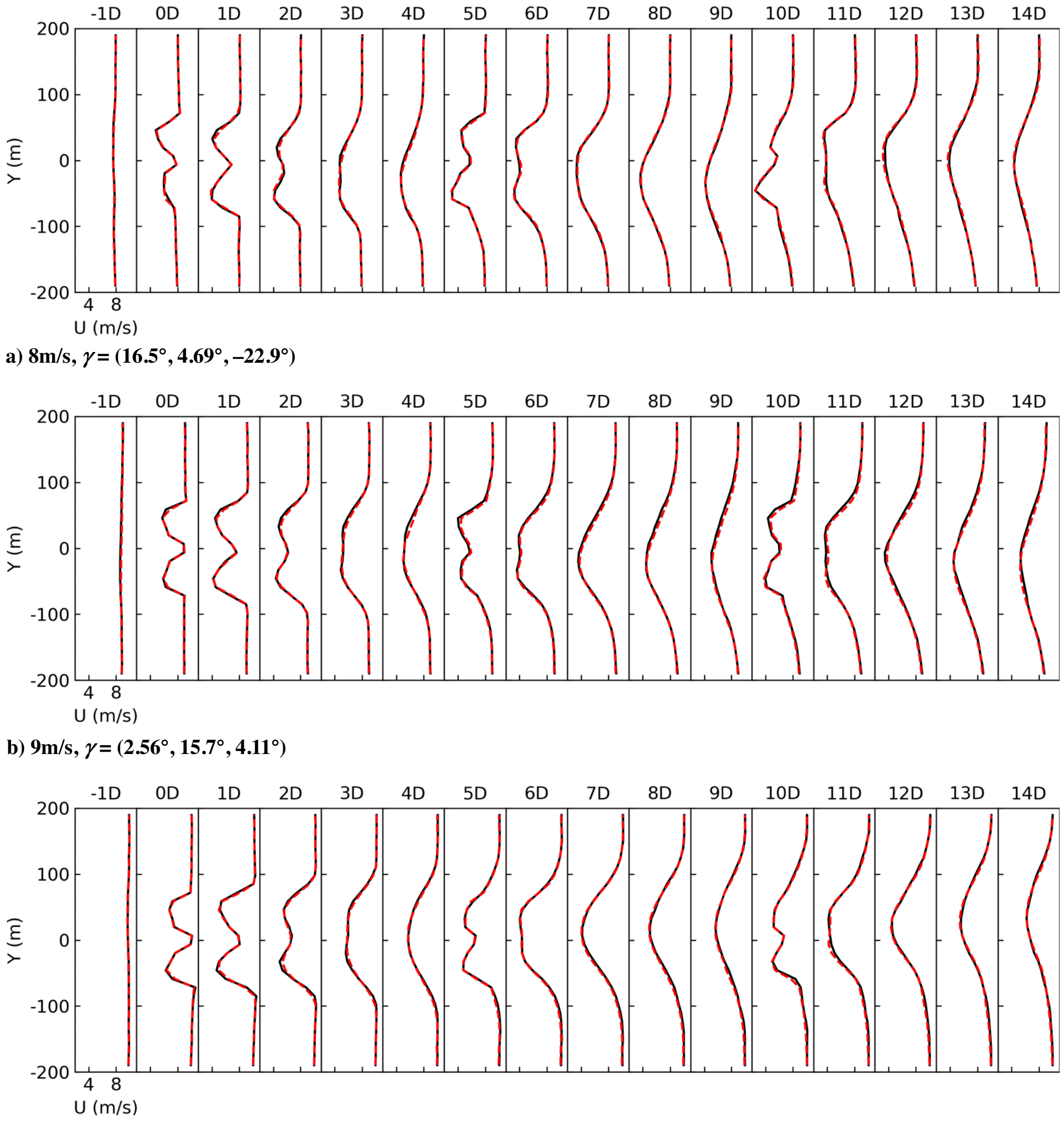
c) 10m/s, $\gamma = (-6.98°, -15.4°, -23.4°)$

Fig. 13 The comparisons between ICA-NN predictions and high-fidelity simulations for three-turbine cases.

At the same time, the wake expands in the spanwise direction as the wake deficit decreases. As for the case shown in Fig. 12b, the inflow at $X = -1D$ is of Gaussian shape, which is generated by the upstream turbines; and the wake development from $X = 0D$ to $X = 4D$ shows similar features as in the freestream case. In summary, the velocity profile predictions match perfectly with the high-fidelity simulation results at all streamwise locations; and all the main flow features (such as the wake recovery, the wake deflection, and the wake expansion) are captured very well by the surrogate model under various turbine yaw and inflow conditions. As ICA-NN is the best among all three methods in terms of the prediction error, as shown in Table 2, it is used in the following for the surrogate modeling of wind turbine arrays.

The prediction of the flowfield around distributed wind turbines follows from Algorithm 2. As the problem here is convection dominant, the iterative updating process in Algorithm 2 is ignored and the flow quantities at the subdomain boundary are directly imposed according to the upstream surrogate model predictions. Specifically, the turbines' relative positions are first determined according to the wind direction and turbine coordinates. Then, the wind fields around the front turbines are predicted by a surrogate model with their yaw angles and the freestream inflow profile as the model input. Next, the flowfields around subsequent turbines are predicted similarly, but with the inflow profile extracted from the surrogate model prediction of the corresponding upstream subdomains. In this way, the flowfield in the whole wind farm can be obtained.

The results of a wind farm with three turbines in a row are given here to test the surrogate model's accuracy, efficiency, and scalability. Twenty samples of turbine yaw angles are generated, and the SOWFA is employed to generate the high-fidelity test data in order to assess the accuracy of the surrogate model. The surrogate model predictions and the SOWFA results are only based on the same distributed yaw angles and the same freestream inflow profile. The results of three typical test cases are given in Fig. 13, where the mean freestream wind speeds are 8, 9, and 10 m/s, respectively. As shown, the prediction errors are quite small in the whole domain for all the test cases. To further examine the surrogate model's performance, the corresponding velocity profiles at various locations from $X = -1D$ to $X = 14D$ are given in Fig. 14. As can be seen, the wake deficits generated by the turbines at $X = 0D$, $X = 5D$, and $X = 10D$; the wake development including deflection, recovery, and expansion behind each turbine; and the wake interactions between turbines are all successfully predicted. The predicted profiles at all the locations match very well with the corresponding high-fidelity simulation results. Also, the average RMSE of the surrogate model predictions compared to the SOWFA results for all 20 test cases is 1.60×10^{-1} m/s, which is just 2% of the freestream wind speed; and the online run time of the surrogate model is negligible (around



a) 8m/s, γ = (16.5°, 4.69°, –22.9°)

b) 9m/s, γ = (2.56°, 15.7°, 4.11°)

c) 10m/s, γ = (–6.98°, –15.4°, –23.4°)

Fig. 14 The velocity profiles for the three-turbine cases by ICA-NN (dashed) and high-fidelity simulations (solid).

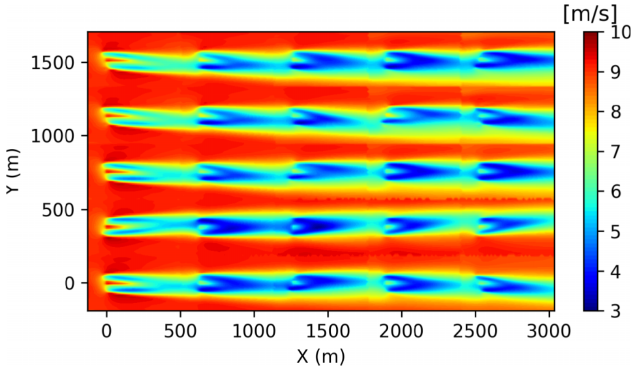


Fig. 15 The wake prediction of a wind farm with 5 × 5 wind turbines.

9×10^{-4} s using one core) compared to the SOWFA, which requires around 44 h using 256 cores for each case. Therefore, it is concluded that the surrogate model predicts the wind farm flowfield efficiently and accurately.

To further illustrate the scalability of the surrogate model, the prediction of a wind farm with 5×5 wind turbines is carried out; and the results are given in Fig. 15. The inflow velocity is set as 9 m/s, and the turbine yaw angles from left to right are set as [28.4, 13.3, −7.63, −8.50, −28.1 deg] for the first row, [27.9, 8.50, 21.5, −29.4, 25.1 deg] for the second row, [19.5, 16.5, −26.5, −9.15, −10.8 deg] for the third row, [−15.7, 6.54, −12.1, −14.6, −16.9 deg] for the fourth row, and [16.7, −2.79, −27.9, 10.8, 26.9 deg] for the fifth row. As can be seen, the wake interactions and the yaw effects are both captured satisfactorily by the surrogate model. Because the so-constructed surrogate model can achieve fast approximation of the original high-fidelity LES model, it can be used directly for the control design of large-scale wind farms.

IV. Conclusions

In this paper, a machine-learning-based surrogate modeling method was proposed, where three-dimensionality reduction techniques (i.e., POD, ICA, and AE) were investigated for reducing the flowfield dimension; and the neural network was used to predict the reduced coefficients from the input parameters. The surrogate modeling method was specifically designed to tackle distributed fluid systems by carrying out surrogate modeling for each subdomain and combining the flowfield of each subdomain with the consideration of the matching condition at the interface. The applications to a diffusion-dominant problem (more specifically, 1-D Poisson equation) and a convection-dominant problem (more specifically, wind farm simulations) demonstrated the efficiency, accuracy, and scalability of the proposed surrogate modeling method. In particular, the surrogate model of wind farm wakes predicted the wind farm velocity field very accurately, with an average root-mean-square error (compared to high-fidelity results) being 2% of the freestream wind speed; whereas the online prediction time is negligible (around 9×10^{-4} s using one core) compared to the high-fidelity simulation, which requires around 44 h using 256 cores for each case. This demonstrated the ability of the proposed method in modeling large-scale distributed fluid systems.

Future research may involve the application of the proposed surrogate modeling to other flow problems, such as the flowfield around heater arrays in heat exchangers and the distributed roughness elements in boundary-layer flow. The optimal design using the constructed surrogate model is also of great interest. For example, it can be used for wind farm layout optimization or as an internal model for wind farm yaw optimization. Another research direction is the surrogate modeling of unsteady distributed systems and the optimal control design based on such surrogate models.

Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation program under Marie Sklodowska-Curie grant agreement no. 765579. The authors also acknowledge the Scientific Computing Research Technology Platform at the University of Warwick for providing high-performance computing resources.

References

[1] Amsallem, D., Cortial, J., and Farhat, C., "Towards Real-Time Computational-Fluid-Dynamics-Based Aeroelastic Computations Using a Database of Reduced-Order Information," *AIAA Journal*, Vol. 48, No. 9, 2010, pp. 2029–2037.
https://doi.org/10.2514/1.j050233

[2] Fossati, M., "Evaluation of Aerodynamic Loads via Reduced-Order Methodology," *AIAA Journal*, Vol. 53, No. 8, 2015, pp. 2389–2405.
https://doi.org/10.2514/1.j053755

[3] Zhang, J., and Fu, S., "An Efficient Approach for Quantifying Parameter Uncertainty in the SST Turbulence Model," *Computers and Fluids*, Vol. 181, March 2019, pp. 173–187.
https://doi.org/10.1016/j.compfluid.2019.01.017

[4] Wang, Q., Hesthaven, J. S., and Ray, D., "Non-Intrusive Reduced Order Modeling of Unsteady Flows Using Artificial Neural Networks with Application to a Combustion Problem," *Journal of Computational Physics*, Vol. 384, May 2019, pp. 289–307.
https://doi.org/10.1016/j.jcp.2019.01.031

[5] Berkooz, G., Holmes, P., and Lumley, J. L., "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows," *Annual Review of Fluid Mechanics*, Vol. 25, No. 1, 1993, pp. 539–575.
https://doi.org/10.1146/annurev.fl.25.010193.002543

[6] Chatterjee, A., "An Introduction to the Proper Orthogonal Decomposition," *Current Science*, Vol. 78, No. 7, 2000, pp. 808–817.

[7] Liang, Y., Lee, H., Lim, S., Lin, W., Lee, K., and Wu, C., "Proper Orthogonal Decomposition and Its Applications—Part I: Theory," *Journal of Sound and Vibration*, Vol. 252, No. 3, 2002, pp. 527–544.
https://doi.org/10.1006/jsvi.2001.4041

[8] Schmid, P. J., "Dynamic Mode Decomposition of Numerical and Experimental Data," *Journal of Fluid Mechanics*, Vol. 656, Aug. 2010, pp. 5–28.
https://doi.org/10.1017/s0022112010001217

[9] Dupuis, R., Jouhaud, J.-C., and Sagaut, P., "Surrogate Modeling of Aerodynamic Simulations for Multiple Operating Conditions Using Machine Learning," *AIAA Journal*, Vol. 56, No. 9, 2018, pp. 3622–3635.
https://doi.org/10.2514/1.j056405

[10] Swischuk, R., Kramer, B., Huang, C., and Willcox, K., "Learning Physics-Based Reduced-Order Models for a Single-Injector Combustion Process," *AIAA Journal*, Vol. 58, No. 6, 2020, pp. 2658–2672.
https://doi.org/10.2514/1.j058943

[11] Guo, X., Li, W., and Iorio, F., "Convolutional Neural Networks for Steady Flow Approximation," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM Press, New York, 2016, pp. 481–490.
https://doi.org/10.1145/2939672.2939738

[12] Jin, X., Cheng, P., Chen, W.-L., and Li, H., "Prediction Model of Velocity Field Around Circular Cylinder over Various Reynolds Numbers by Fusion Convolutional Neural Networks Based on Pressure on the Cylinder," *Physics of Fluids*, Vol. 30, No. 4, 2018, Paper 047105.
https://doi.org/10.1063/1.5024595

[13] Hesthaven, J., and Ubbiali, S., "Non-Intrusive Reduced Order Modeling of Nonlinear Problems Using Neural Networks," *Journal of Computational Physics*, Vol. 363, June 2018, pp. 55–78.
https://doi.org/10.1016/j.jcp.2018.02.037

[14] Guo, M., and Hesthaven, J. S., "Reduced Order Modeling for Nonlinear Structural Analysis Using Gaussian Process Regression," *Computer Methods in Applied Mechanics and Engineering*, Vol. 341, Nov. 2018, pp. 807–826.
https://doi.org/10.1016/j.cma.2018.07.017

[15] Guo, M., and Hesthaven, J. S., "Data-Driven Reduced Order Modeling for Time-Dependent Problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 345, March 2019, pp. 75–99.
https://doi.org/10.1016/j.cma.2018.10.029

[16] Swischuk, R., Mainini, L., Peherstorfer, B., and Willcox, K., "Projection-Based Model Reduction: Formulations for Physics-Based Machine Learning," *Computers and Fluids*, Vol. 179, Jan. 2019, pp. 704–717.
https://doi.org/10.1016/j.compfluid.2018.07.021

[17] Hyvärinen, A., Karhunen, J., and Oja, E., *Independent Component Analysis*, Wiley, New York, 2001.
https://doi.org/10.1002/0471221317

[18] Hinton, G. E., "Reducing the Dimensionality of Data with Neural Networks," *Science*, Vol. 313, No. 5786, 2006, pp. 504–507.
https://doi.org/10.1126/science.1127647

[19] Tou, S., Tso, C., and Zhang, X., "3-D Numerical Analysis of Natural Convective Liquid Cooling of a 3 × 3 Heater Array in Rectangular

Enclosures," *International Journal of Heat and Mass Transfer*, Vol. 42, No. 17, 1999, pp. 3231–3244. https://doi.org/10.1016/s0017-9310(98)00379-2

[20] Mon, M. S., and Gross, U., "Numerical Study of Fin-Spacing Effects in Annular-Finned Tube Heat Exchangers," *International Journal of Heat and Mass Transfer*, Vol. 47, Nos. 8–9, 2004, pp. 1953–1964. https://doi.org/10.1016/j.ijheatmasstransfer.2003.09.034

[21] Rizzetta, D. P., and Visbal, M. R., "Direct Numerical Simulations of Flow Past an Array of Distributed Roughness Elements," *AIAA Journal*, Vol. 45, No. 8, 2007, pp. 1967–1976. https://doi.org/10.2514/1.25916

[22] Liu, J., Srebric, J., and Yu, N., "Numerical Simulation of Convective Heat Transfer Coefficients at the External Surfaces of Building Arrays Immersed in a Turbulent Boundary Layer," *International Journal of Heat and Mass Transfer*, Vol. 61, June 2013, pp. 209–225. https://doi.org/10.1016/j.ijheatmasstransfer.2013.02.005

[23] Calaf, M., Meneveau, C., and Meyers, J., "Large Eddy Simulation Study of Fully Developed Wind-Turbine Array Boundary Layers," *Physics of Fluids*, Vol. 22, No. 1, 2010, Paper 015110. https://doi.org/10.1063/1.3291077

[24] Wu, Y.-T., and Porté-Agel, F., "Modeling Turbine Wakes and Power Losses Within a Wind Farm Using LES: An Application to the Horns Rev Offshore Wind Farm," *Renewable Energy*, Vol. 75, March 2015, pp. 945–955. https://doi.org/10.1016/j.renene.2014.06.019

[25] Nilsson, K., Ivanell, S., Hansen, K. S., Mikkelsen, R., Sørensen, J. N., Breton, S.-P., and Henningson, D., "Large-Eddy Simulations of the Lillgrund Wind Farm," *Wind Energy*, Vol. 18, No. 3, 2014, pp. 449–467. https://doi.org/10.1002/we.1707

[26] "SOWFA," National Wind Technology Center's Information Portal (online database), U.S. Dept. of Energy, National Renewable Energy Laboratory, 2014, https://nwtc.nrel.gov/SOWFA [retrieved 8 Feb. 2019].

[27] Boersma, S., Doekemeijer, B., Gebraad, P., Fleming, P., Annoni, J., Scholbrock, A., Frederik, J., and van Wingerden, J.-W., "A Tutorial on Control-Oriented Modeling and Control of Wind Farms," *2017 American Control Conference (ACC)*, Inst. of Electrical and Electronics Engineers, New York, 2017, pp. 1–18. https://doi.org/10.23919/acc.2017.7962923

[28] Meyers, J., and Meneveau, C., "Large Eddy Simulations of Large Wind-Turbine Arrays in the Atmospheric Boundary Layer," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2010-0827, 2010. https://doi.org/10.2514/6.2010-827

[29] Churchfield, M. J., Lee, S., Michalakes, J., and Moriarty, P. J., "A Numerical Study of the Effects of Atmospheric and Wake Turbulence on Wind Turbine Dynamics," *Journal of Turbulence*, Vol. 13, Jan. 2012, Paper N14. https://doi.org/10.1080/14685248.2012.668191

[30] Witha, B., Steinfeld, G., and Heinemann, D., "High-Resolution Offshore Wake Simulations with the LES Model PALM," *Research Topics in Wind Energy*, Springer, Berlin, 2014, pp. 175–181. https://doi.org/10.1007/978-3-642-54696-9_26

[31] Martnez-Tossas, L. A., Churchfield, M. J., and Leonardi, S., "Large Eddy Simulations of the Flow Past Wind Turbines: Actuator Line and Disk Modeling," *Wind Energy*, Vol. 18, No. 6, 2014, pp. 1047–1060. https://doi.org/10.1002/we.1747

[32] Jensen, N. O., "A Note on Wind Generator Interaction," RisøNational Lab., Risø-M, No. 2411, Roskilde, Denmark, 1983.

[33] Katic, I., Højstrup, J., and Jensen, N. O., "A Simple Model for Cluster Efficiency," *European Wind Energy Association Conference and Exhibition*, EWEC'86 Proceedings, A. Raguzzi, Vol. 1, Rome, Italy, 1987, pp. 407–410.

[34] Frandsen, S., Barthelmie, R., Pryor, S., Rathmann, O., Larsen, S., Højstrup, J., and Thøgersen, M., "Analytical Modelling of Wind Speed Deficit in Large Offshore Wind Farms," *Wind Energy*, Vol. 9, Nos. 1–2, 2006, pp. 39–53. https://doi.org/10.1002/we.189

[35] Gebraad, P. M. O., Teeuwisse, F. W., van Wingerden, J. W., Fleming, P. A., Ruben, S. D., Marden, J. R., and Pao, L. Y., "Wind Plant Power Optimization Through Yaw Control Using a Parametric Model for Wake Effects—A CFD Simulation Study," *Wind Energy*, Vol. 19, No. 1, 2014, pp. 95–114. https://doi.org/10.1002/we.1822

[36] Niayifar, A., and Porté-Agel, F., "Analytical Modeling of Wind Farms: A New Approach for Power Prediction," *Energies*, Vol. 9, No. 9, 2016, Paper 741. https://doi.org/10.3390/en9090741

[37] Iungo, G. V., Santoni-Ortiz, C., Abkar, M., Porté-Agel, F., Rotea, M. A., and Leonardi, S., "Data-Driven Reduced Order Model for Prediction of Wind Turbine Wakes," *Journal of Physics: Conference Series*, Vol. 625, No. 1, 2015, Paper 012009. https://doi.org/10.1088/1742-6596/625/1/012009

[38] Hamilton, N., Viggiano, B., Calaf, M., Tutkun, M., and Cal, R. B., "A Generalized Framework for Reduced-Order Modeling of a Wind Turbine Wake," *Wind Energy*, Vol. 21, No. 6, 2018, pp. 373–390. https://doi.org/10.1002/we.2167

[39] Annoni, J., Gebraad, P., and Seiler, P., "Wind Farm Flow Modeling Using an Input-Output Reduced-Order Model," *2016 American Control Conference (ACC)*, Inst. of Electrical and Electronics Engineers, New York, 2016, pp. 506–512. https://doi.org/10.1109/acc.2016.7524964

[40] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-Learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, Nov. 2011, pp. 2825–2830. https://dl.acm.org/doi/10.5555/1953048.2078195

[41] Chollet, F., et al., Keras (online database), 2015, https://github.com/fchollet/keras [retrieved 17 May 2019].

[42] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," Preprint submitted 22 Dec. 2014, https://arxiv.org/abs/1412.6980.

[43] Hyvärinen, A., and Oja, E., "Independent Component Analysis: Algorithms and Applications," *Neural Networks*, Vol. 13, Nos. 4–5, 2000, pp. 411–430. https://doi.org/10.1016/s0893-6080(00)00026-5

[44] Churchfield, M., Lee, S., Moriarty, P., Martinez, L., Leonardi, S., Vijayakumar, G., and Brasseur, J., "A Large-Eddy Simulations of Wind-Plant Aerodynamics," *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2012-0537, 2012. https://doi.org/10.2514/6.2012-537

[45] Fleming, P., Gebraad, P. M., Lee, S., van Wingerden, J.-W., Johnson, K., Churchfield, M., Michalakes, J., Spalart, P., and Moriarty, P., "Simulation Comparison of Wake Mitigation Control Strategies for a Two-Turbine Case," *Wind Energy*, Vol. 18, No. 12, 2014, pp. 2135–2143. https://doi.org/10.1002/we.1810

R. Ghanem
Associate Editor