Predictive large-eddy-simulation wall modeling via physics-informed neural networks

X. I. A. Yang,[1,*] S. Zafar,[1] J.-X. Wang,[2,3] and H. Xiao[4]

[1]*Department of Mechanical Engineering, Penn State University, University Park, Pennsylvania 16802, USA*

[2]*Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, Indiana 46556, USA*

[3]*Center for Informatics and Computational Science, University of Notre Dame, Notre Dame, Indiana 46556, USA*

[4]*Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, Virginia 24061, USA*

(Received 11 July 2018; published 15 March 2019)

While data-based approaches were found to be useful for subgrid scale (SGS) modeling in Reynolds-averaged Navier-Stokes (RANS) simulations, there have not been many attempts at using machine learning techniques for wall modeling in large-eddy simulations (LESs). Large-eddy simulation differs from RANS simulation in many aspects. For one thing, LES is scale resolving. For another, LES is in and of itself a high-fidelity tool. Because data sets of higher fidelity are in general not frequently accessible or available, this poses additional challenges to data-based modeling in LES. Further, SGS modeling usually needs flow information at only large scales, in contrast with wall modeling, which needs to account for both near-wall small scales and large scales above the wall. In this work we discuss how the above-noted challenges may be addressed when taking a data-based approach for wall modeling. We also show the necessity of incorporating physical insights in model inputs, i.e., using inputs that are inspired by the vertically integrated thin-boundary-layer equations and the eddy population density scalings. We show that the inclusion of the above physics-based considerations would enhance extrapolation capabilities of a neural network to flow conditions that are not within the train data. Being cheap to evaluate and using only channel flow data at $\mathrm{Re}_\tau = 1000$, the trained networks are found to capture the law of the wall at arbitrary Reynolds numbers and outperform the conventional equilibrium model in a nonequilibrium flow.

DOI: 10.1103/PhysRevFluids.4.034602

I. INTRODUCTION

Wall-bounded flows are often encountered in real-world engineering [1–3]. Resolving all scales and conducting direct numerical simulations (DNSs) for engineering analysis is usually not possible because of limited computational resources and the high Reynolds numbers of practically relevant flows [4–6]. The same is true for wall-resolved large-eddy simulation (LES), where large-scale energy-containing motions are resolved. To alleviate the strict near-wall resolution requirement and reduce computational cost, wall models are often necessary [7]. In the case of incompressible flows, for instance, an LES wall model is a model for wall-shear stress. Simple as it may sound in terms of purpose, the fact remains that wall modeling is the pacing item in LES [8,9]. Conventionally, LES wall models have been physics and mathematics based [10,11]. Here we briefly review a few of the

*Corresponding author: xiangyang@psu.edu

existing wall modeling techniques. The commonly used algebraic equilibrium wall model specifies wall-shear stress by enforcing the law of the wall locally and instantaneously [12]. The zonal model solves the thin-boundary-layer equation on a set of refined mesh near the wall [13–15]. The integral wall model adds to the otherwise equilibrium logarithmic velocity profile an additional linear term and accounts for near-wall nonequilibrium effects by solving the vertically integrated momentum equation [16]. A similar approach was pursued in Ref. [17], where a plug flow is used within the wall-adjacent cell. The wall model by Inoue *et al.* [18] models wall-shear stress fluctuations using the predictive inner-outer model [2,19]. Finally, the dynamic slip wall model models the wall-shear stress by applying a differential filter in LES [20,21].

The conventional wall models as reviewed above have been quite useful in predictive modeling for both low- and high-speed flows [22–28]. However, with an increasing number of high-fidelity DNS data sets being made publicly available [29–34], augmenting physics-based models with data-based approaches or vice versa may aid researchers in solving some of the pressing issues in wall modeled LES (WMLES), including, but not limited to, flow separation and heat transfer [8]. Data-based approaches, particularly machine learning (ML), are conventionally used for video classification [35] and voice recognition [36]. Recently, machine learning techniques such as neural networks and random forests were used for subgrid scale (SGS) modeling in Reynolds-averaged Navier-Stokes (RANS) simulations [37–39] and there have been a few works utilizing physics-informed data-based approaches [40–44]. For example, Ling *et al.* [40,41] embedded tensorial invariance in network input and Xiao and co-workers [43,44] developed physics-informed random forests where a number of relevant flow quantities with physics-based normalizations are fed to the model. The reader is directed to the review by Duraisamy *et al.* [45] for a detailed overview of recent advancements. While data-based approaches are advantageous for applications such as SGS stress and scalar flux modeling [46–49], wall-bounded flows [50,51], and multiphase flow turbulence model development [52,53], in this work ML is used for wall modeling in LES.

Although both wall and SGS models deal with unresolved turbulence, due to the difference in flow physics, modeling Reynolds and wall-shear stresses requires the use of substantially different techniques [54]. Compared to SGS modeling in RANS simulations, WMLES poses additional challenges to data-based approaches. For one, time averaging in (steady) RANS simulation kills turbulence and therefore (steady) RANS simulation is a model for a deterministic system. Large-eddy simulation, on the other hand, is scale resolving and therefore models a stochastic system. This difference is crucial since the training process in ML for modeling a stochastic system is inherently more difficult than modeling a deterministic system. This difficulty is exacerbated by the necessity of more training data for a stochastic system. The second challenge is also physics related. The purpose of SGS models is to dissipate turbulent kinetic energy at large scales so that an LES grid does not need to resolve the Kolmogorov scales [55]. Turbulence dissipation is only a function of large scales, therefore calibrating an SGS model at moderate Reynolds numbers and using the calibrated model for flows at high Reynolds numbers are usually not problematic [56]. The same cannot be said for wall modeling, however. This is because wall-shear stress depends on both large scales and small scales, e.g., the friction would be different on a smooth versus rough wall even if the large-scale motions above the wall are the same.

The starting point for using an ML technique is often a lack of knowledge of the physics that is needed for developing a closure model. To that end, one often needs high-fidelity data to train a closure model for low-fidelity simulations. Both LES and DNS are high fidelity for RANS simulation, therefore data obtained from either of the two or from experimental measurements can be used for data-based RANS modeling. Moreover, because only mean fields such as mean velocities, Reynolds stresses, or turbulent budget terms are needed for the development of data-based closure models for RANS simulations, data sharing is often not difficult, and several such databases are available [57–59]. For LES, however, only DNS is high fidelity, and due to prohibitive computational costs of DNS and difficulties in data sharing, high-fidelity DNS data sets are often unavailable for use in data-based modeling. While the database at Johns Hopkins University has been a great platform for sharing high-fidelity DNS data and for leveraging the

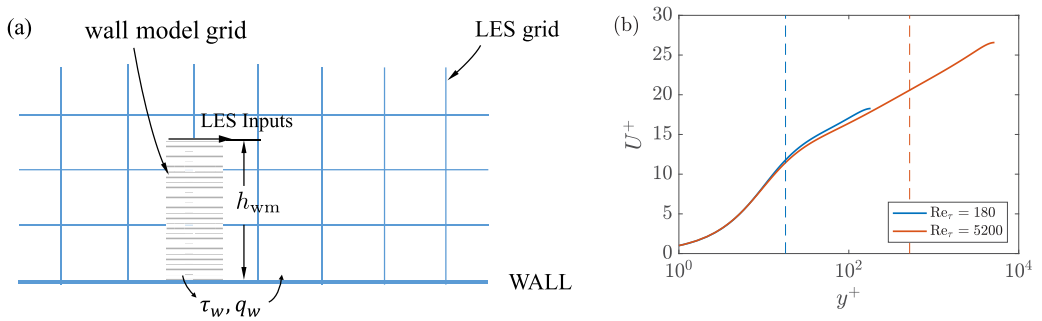

FIG. 1. (a) Schematic of the setup of WMLES. The near-wall turbulence is not resolved, therefore the no-slip condition does not apply. Instead, a wall model is used. (b) Mean velocity profiles of a channel at $\mathrm{Re}_\tau \approx 180$ [64] and $\mathrm{Re}_\tau \approx 5200$ [31]. The plus superscript indicates normalization by wall units. The dashed lines are at $y^+ = 18$ and $y^+ = 520$, respectively, which are roughly the locations of the first off-wall grid of typical WMLES of the two flows.

shared data for developing LES closure models [30,46,48,60,61], the available DNS data are limited to canonical flows in simple geometries, including only channel flow, boundary-layer flow, and isotropic turbulence. The challenge in data sharing is possibly an explanation for why wall modeling has traditionally relied on physics-based approaches. If data-based approaches are to be useful for WMLES, ML must only be done using limited data, but the model needs to work for all flows. This is also the most oft-noted difficulty of ML, i.e., the ability to extrapolate, also known as generalization.

The traditional ML approaches have been solely based on data, and that may not be suitable for the WMLES. To address this issue, a physics-informed ML approach [62,63], where physics-based constraints and knowledge are imposed within the learning process, is preferred. Leveraging the known physical laws and constraints would compensate for the lack of high-fidelity training data and achieve more efficient learning.

For WMLES, we prefer the physics-informed approach. Here we briefly discuss why. In WMLES, wall-shear stress is modeled using resolved LES velocity at a distance h_{wm} from the wall, where h_{wm} is the distance of the first (dependent on the model implementation, second or even the third [15]) off-wall grid point from the wall. Figure 1(a) shows a schematic of the typical setup of a WMLES, where about $O(10)$ grid points are used to resolve the flow in one boundary-layer height and the wall-shear stress is modeled according to a wall model. Figure 1(b) shows the mean velocity profiles in channel flow at $\mathrm{Re}_\tau \approx 180$ [64] and $\mathrm{Re}_\tau \approx 5200$ [31], where $\mathrm{Re}_\tau = u_\tau \delta/\nu$ is the friction Reynolds number, $u_\tau = \sqrt{\tau_w/\rho}$ is the friction velocity, τ_w is the mean wall-shear stress, $\rho = \mathrm{const}$ is the fluid density, δ is the half channel height, and ν is the kinematic viscosity. Without incorporating any physics, a neural network trained using data at $\mathrm{Re}_\tau \approx 180$ probably cannot be directly used for flow at $\mathrm{Re}_\tau \approx 5200$, where the first off-wall grid point of the $\mathrm{Re}_\tau \approx 5200$ LES, which is located at $h_{\mathrm{wm}} \approx 520$, would already be above 180.

That is to say, flow at $\mathrm{Re}_\tau \approx 180$ would not provide any useful information for data-based modeling of flow at $\mathrm{Re}_\tau \approx 5200$ if no physical insight is incorporated, simply because the entire flow of the $\mathrm{Re}_\tau \approx 180$ channel is within the first grid of the $\mathrm{Re}_\tau \approx 5200$ LES. Unfortunately, while one can include the $\mathrm{Re}_\tau = 5200$ data in the training, real-world problems that require the use of WMLES for engineering design involve flows at higher Reynolds numbers, for which often neither DNS nor experimental measurements are available. Hence, we prefer physics-informed ML approaches for WMLES, which, as we will show, have the extrapolating capability we need.

The rest of the paper is organized as follows. We briefly recap background information in Sec. II. Two model problems are considered in Sec. III. In Sec. IV we present results of neural networks for WMLES. We briefly discuss the present approach in Sec. V. Concluding remarks are given in Sec. VI.

II. BACKGROUND

In this section we provide a summary of relevant background information and the reasoning for a few parameter choices in Secs. III and IV.

A. Flow physics in wall-modeled LES and choice of input features for learning

The law of the wall (LOW) reads

$$\langle u^+ \rangle = \frac{1}{\kappa} \ln\left(\frac{y}{y_0}\right), \tag{1}$$

where $\langle\cdot\rangle$ is the ensemble average of the bracketed quantity, $\kappa \approx 0.4$ is the von Kármán constant, $y_0 = \nu/u_\tau \exp(-\kappa B)$ is a viscous scale, $B \approx 5$ is another constant, and ln is the natural logarithm. The mean flow follows the LOW in the logarithmic range $30 \lesssim y^+$ and $y/\delta \lesssim 0.15$ [65]. Within the logarithmic range, the flow may be considered as universal [66]. A wall model can and should model only the universal near-wall flow behavior.

Flow behavior near a solid boundary is, by and large, governed by the thin-boundary-layer equation

$$\frac{\partial u}{\partial t} + \frac{\partial u u_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu \frac{\partial^2 u}{\partial y \partial y}, \tag{2}$$

where u_i ($i = 1, 2, 3$) is the velocity in the ith Cartesian direction; u is the streamwise velocity; u, v, and w are interchangeably used for u_1, u_2, and u_3, respectively; x, y, and z are the streamwise, wall-normal, and spanwise directions, respectively, and are interchangeably used with x_i ($i = 1, 2, 3$); t is the time; and p is the pressure. The solution to Eq. (2) is, in principle, a good approximation of near-wall flow. However, solving Eq. (2) is often costly [13,25], and a more cost-effective way is to solve the vertically integrated equation [16]. Time averaging Eq. (2) and integrating from the wall to a wall normal distance $h_{\rm wm}$ leads to

$$\langle \tau_w \rangle = \nu \frac{\partial \langle u \rangle}{\partial y}\bigg|_{y=0} = -\langle uv \rangle|_{h_{\rm wm}} - \int_0^{h_{\rm wm}} \frac{1}{\rho}\frac{\partial \langle p \rangle}{\partial x} dy + \nu \frac{\partial \langle u \rangle}{\partial y}\bigg|_{h_{\rm wm}} - \int_0^{h_{\rm wm}} \frac{\partial \langle uu \rangle}{\partial x} + \frac{\partial \langle uw \rangle}{\partial z} dy, \tag{3}$$

where the unsteady term is dropped and the wall-shear stress depends on [from left to right in Eq. (3)] the Reynolds stress $-\langle uv \rangle$, the pressure gradient term, the viscous term, and the convective term. Although all the terms are functions of primitive flow quantities and one may use primitive flow quantities u, v, w, $h_{\rm wm}$, p, etc., to model τ_w, according to Eq. (3), it is more straightforward to use rearranged nondimensional LES quantities including $(\partial p/\partial x)h_{\rm wm}$ and $uv|_{h_{\rm wm}}$ as model input for better predictive generality [16,67]. Ideally, a network will use all available flow information as input, determine what information is relevant in a particular context, and output the wall-shear stress.

Including all the terms in a wall model is usually not straightforward, and it is quite common that a wall model models only the term $-\langle uv \rangle$ using only $u_\parallel$ and $h_{\rm wm}$ as model input, where $u_\parallel$ is the wall-parallel velocity at a distance $h_{\rm wm}$ from the wall. If one uses only $u_\parallel$ and $h_{\rm wm}$ as model input, one needs to determine the exact form of the input. For example, one may directly use $h_{\rm wm}^+$ and $u_\parallel^+$ as the input. Alternatively, one may use $f_2(u_\parallel, h_{\rm wm})$ and $f_1(u_\parallel, h_{\rm wm})$ as the input, as long as it is possible to back out both $u_\parallel$ and $h_{\rm wm}$ from $f_1(u_\parallel, h_{\rm wm})$ and $f_2(u_\parallel, h_{\rm wm})$. Here f_1 and f_2 are two generic functions. Although any independent $f_1(u_\parallel, h_{\rm wm})$ and $f_2(u_\parallel, h_{\rm wm})$ may be used, selecting a set of input that has the right physics reduces training efforts and improves model accuracy [40,41].

In this work, such physical insights will be provided by the hierarchical random additive process model [68–72]. Figure 2 shows a schematic of the modeled boundary-layer flow, where high-Reynolds-number boundary layers are modeled as collections of space-filling, wall-attached, self-similar eddies. The population density of the attached eddies is $\mathcal{P} \sim 1/y$. If the friction velocity $u_\tau = \sqrt{\tau_w/\rho}$ is unity, i.e., using u_τ for normalization, the velocity at a wall-normal distance $h_{\rm wm}$ is proportional to the number of attached eddies at that height. The number of attached eddies at

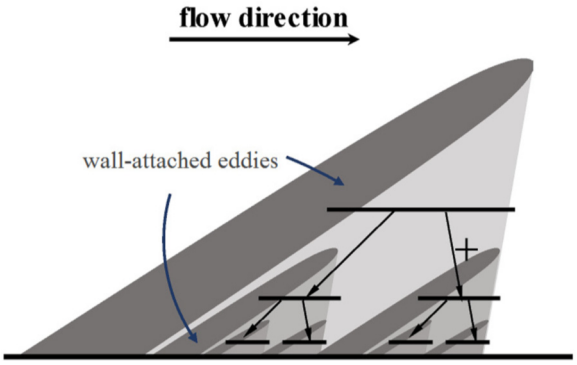

FIG. 2. Schematic of the modeled boundary-layer structure. The attached eddies are space filling and therefore the number of eddies doubles as their sizes halve.

a wall-normal distance $h_{\rm wm}$ in the logarithmic range is approximately $N = \int \mathcal{P}(y)dy \sim \ln(h_{\rm wm})$; this number is $N \sim h_{\rm wm}$ in the viscous sublayer. Hence, $\ln(h_{\rm wm}/y_0)/u_\parallel^+$ and $u_\parallel^+/h^+$ are good approximation of wall-shear stress for $h_{\rm wm}$ being in the logarithmic range and the viscous sublayer. In the following we will use $\ln(h_{\rm wm}/y_0)/u_\parallel^+$ and $u_\parallel^+/h^+$ as model input and we will show first that both $\ln(h_{\rm wm}/y_0)/u_\parallel^+$ and $u_\parallel^+/h^+$ are only weak functions of the Reynolds number and second that the trained physics-informed model would be able to be used at any Reynolds numbers.

The discussion above focused on using velocity and wall-normal distance as input and how one may include physical insights. Both flow acceleration and convection can be included as model input and while one may use directly primitive flow quantities including p, du/dx, du/dz, etc., using the integrals in Eq. (3) is simply more physical and more convenient.

B. Neural network

We will use a feedforward neural network architecture for LES wall modeling, Figure 3 shows a schematic of a feedforward neural network. A feedforward neural network contains one input layer, multiple hidden layers, and one output layer. The outputs of each layer are fed forward as inputs to the next layer, with the input layer being the starting point and the output layer being the model prediction. Each layer has a number of neurons, which are computational units that take weighted sums of the inputs to an activation function. In general, we prefer not to use primitive physical quantities, i.e., velocities and pressure, as inputs to a neural net. Previous studies [43,44,73] investigated the choice of ML inputs. Their conclusion was that inputs for ML need to be Galilean invariance, rotation invariance, and properly normalized. However, for wall modeling,

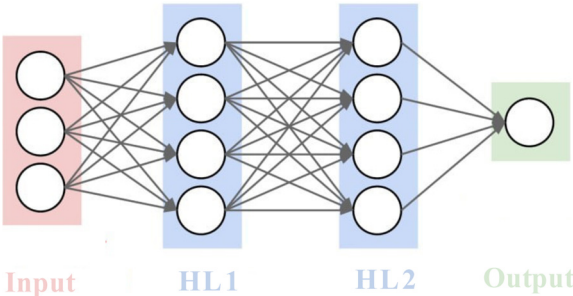

FIG. 3. Schematic of a sample neural network. The neural network is fully connected. It has two hidden layers HL1 and HL2. The input layer contains three neurons and the output layer contains one neuron.

feeding velocity and pressure directly to a network may be acceptable because the wall provides an intrinsic velocity reference and a local coordinate system.

The weights and bias of each neuron in the network are parameters that are determined through fitting a set of training data utilizing backpropagation and an optimization algorithm such as the stochastic gradient descent algorithm [74] and the Levenberg-Marquardt algorithm [75]. The fitting procedure minimizes an objective function, which is usually a measure of misfit between model and data, i.e., the error of the trained model. To prevent overfitting and improve generalization, early stopping criteria are often used [76]. Training is halted if the model error (objective function) evaluated using the data in the validation set stops decreasing for a few epochs, where in one epoch all weights in a neural network are updated after each training vector is passed through the training algorithm.

III. MODEL PROBLEMS

We consider two model problems in this section and discuss briefly how we may go about addressing the challenges noted in Sec. I.

A. Intermittency

For the first model problem, we model a stochastic system whose behavior is governed by the equation $\tau = u^2(1 + r)$, where u and τ are the input (velocity) and the output (wall stress) of the system, respectively. The input u is a Gaussian random variable with zero mean and unit variance and r is a random number that conforms to a uniform distribution between -0.05 and 0.05. Being Gaussian, u values cluster near its mean. The same behavior is found for velocity fluctuations in a turbulent boundary layer and is known as intermittency. A lack of data at large $|u|$ values compromises learning effectiveness. We discuss a possible solution to this problem from a machine learning perspective.

We train a neural network to model the system, where u is the input and $\tau = f(u)$ is the output. The network contains one hidden layer and the hidden layer contains two neurons. For the activation function, we use the hyperbolic tangent sigmoid transfer function. The training data contain $N = 10^5$ pairs of u and $\tau = f(u)$. The objective function is the weighted mean square of the model error and is

$$e = \frac{1}{N}\sum_{n=1}^{N} w_n[\tau_n - f_{\mathrm{NN}}(u_n)]^2, \tag{4}$$

where N is the total number of training data, $w_n \equiv 1$ is the error weight, and $f_{\mathrm{NN}}(u_n)$ is the network output. The weights and bias in the network are initialized randomly. The neural network models the system as a deterministic one and therefore the expected model behavior is $f_{\mathrm{NN}}(u) = u^2$. The results are shown in Fig. 4. The network [$f_{\mathrm{NN}}(u)$ in Fig. 4] follows the expected quadratic behavior closely for $|u| < 3$, where there is a good number of training data points available, but quickly levels off.

While poor model performance is usually expected for a neural network when conducting extrapolation, the extrapolation capability is highly desirable and even critical for WMLES. For WMLES, training data are only available at low to moderate Reynolds numbers, and a network trained at a low Reynolds number needs to model flow at high Reynolds numbers, where the flow is usually more stochastic, with an increasing number of events away from the mean. Although there is no general solution to this problem, improved model performance is found when assigning error weights w_n according to $w_n = 1/[P(u_n) + \epsilon]$, where $P(u)$ is the probability density function (PDF) of the Gaussian random variable u and $\epsilon = 10^{-5}$ to prevent division by zero. If input probability distributions are not known *a priori*, as is often the case, they can be estimated from data, and once trained, the information on the probability distribution is no longer needed when evaluating the network. The model behavior with errors thus weighted are shown in Fig. 4 and is $f_{\mathrm{NN},w}(u)$. The network follows the expected quadratic behavior up to $x = 4$.

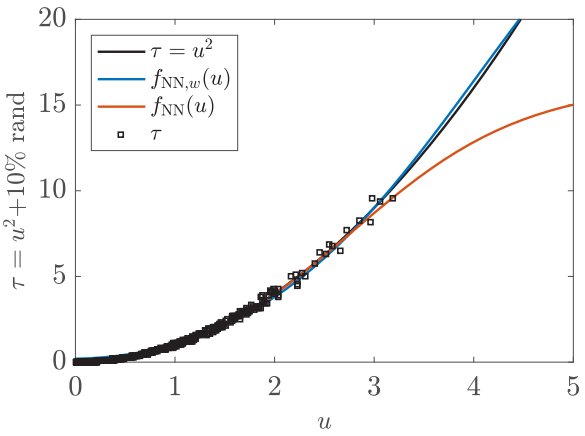

FIG. 4. Training data, network, and expected behavior. Because of symmetry, data are only shown for $x > 0$ for brevity.

The above weighting is such that all u values are weighted evenly: The occurrence of a u value is proportional to the PDF and the weight is inversely proportional to the PDF. Generally, one may weight model error according to $w_n = 1/P^{\alpha}(u_n)$, with $\alpha < 1$, so that the objective function is biased to likely events.

B. Limited training data and aggressive extrapolation

For the second model problem, we train a neural network to model the LOW. We generate data of wall-shear stress and velocity at wall-normal distances $y^+ < 1000$ according to the LOW. The generated data are used for training a neural network and the trained network is then used to predict wall stress given velocities at $y^+ = 10\,000$. The training data are generated as follows. First, we randomly sample 10^5 wall-normal locations between $y^+ = 0$ and $y^+ = 1000$, i.e., h_n, $n = 1, 2, 3, \ldots, 10^5$. Second, the velocities at the sampled wall-normal locations are specified according to the LOW but with 10% of randomness, i.e., $u_n = [1/\kappa \ln(h_n/y_0)](1 + r_n)$, where r_n is a random number sampled from a uniform distribution between -0.1 and 0.1. Finally, we use the equilibrium wall model to compute wall-shear stresses as functions of both the velocity u_n and the wall-normal distance h_n according to

$$\tau^+_{w,n} = h^+_n, \quad y^+ \leqslant 11.3$$
$$\tau^+_{w,n} = \left[\frac{\kappa u^+_n}{\ln(h_n/y_0)}\right]^2, \quad y^+ > 11.3, \tag{5}$$

where all quantities are in wall units. Equation (5) is continuous at $y^+ = 11.3$. The training data contain 10^5 pairs of u_n and h_n as input and τ_w as output. The test data at $y^+ = 10\,000$ are generated in the same manner.

We train two forward-feed networks for this problem. For the first network f_{N1}, we use u^+_n and h^+_n directly as input, i.e., $f_{\mathrm{N1}}(u^+_n, h^+_n)$, and for the second network f_{N2}, we use u^+_n/h^+_n and $\ln(h_n/y_0)/u^+_n$, i.e., $f_{\mathrm{N2}}(u^+_n/h^+_n, \ln(h_n/y_0)/u^+_n)$ (following the discussion in Sec. II A). The behavior of the system to be modeled is slightly more complex than the one in the first model problem and therefore a slightly larger sized network is used. Both networks contain four hidden layers, with ten, ten, ten, and five neurons in the four layers. The above network layout is found to be the best in a simple K-fold cross-validation test, where we split the training set to five groups and tested for five different layouts containing two to six layers. We stop training if the objective function stops decreasing for six epochs (known as early stopping). The training data are generated from uniform distributions

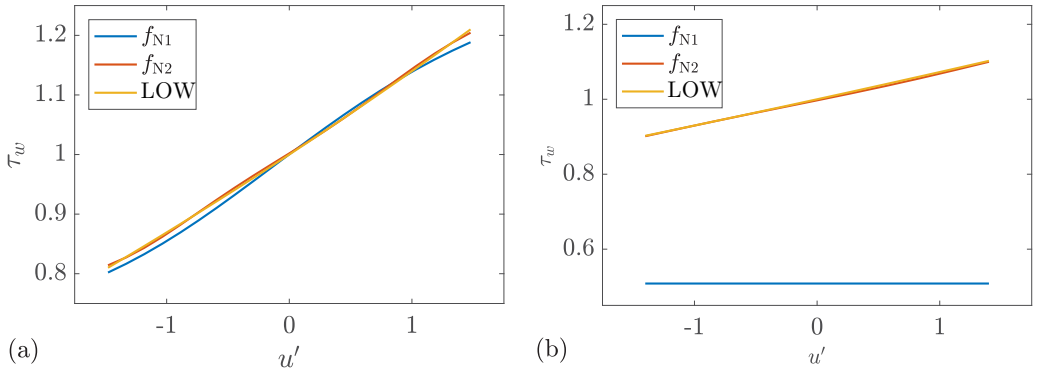

FIG. 5. (a) Wall-shear stress as a function of the velocity at $y^+ = 50$. The mean of the velocity is removed. The LOW is Eq. (5). (b) Same as (a) but at a wall-normal height $y^+ = 10\,000$. In (a), all three lines are almost on top of each other. In (b), f_{N2} and the LOW are on top of each other.

and therefore weighting the errors as discussed in Sec. III A or not does not affect the results. Other details of the two networks are the same as the one in Sec. III A and are not detailed here for brevity.

We first test the model performance at a wall-normal distance $y^+ = 50$, which is well within the wall-normal distance range of the training data. Figure 5(a) shows the model performance and both networks follow the expected LOW closely. However, the two models differ at $y^+ = 10\,000$ [Fig. 5(b)], which is at a wall-normal location beyond the wall-normal distance range of the training data. The net f_{N2} follows the expected LOW behavior. The model prediction is about 50% off for f_{N2}. To briefly summarize, training using regrouped inputs or primitive flow quantities makes no difference if the trained network is used to predict behaviors of a flow in the training data set, but an error of $O(50\%)$, if not higher, is expected if the trained network is used for a flow outside the training data set.

While the data are not real turbulence and are generated according to the LOW, through this exercise, we may still conclude that, at least for WMLES, incorporating the right physics is useful [41,43].

IV. PHYSICS-INFORMED NEURAL NETWORKS

A. Training data

The $\mathrm{Re}_\tau = 1000$ DNS channel flow data [30] are used for training. The computational domain of the DNS is $L_x \times L_y \times L_z = 8\pi\delta \times 2\delta \times 3\pi\delta$ and is resolved by a grid of size $N_x \times N_y \times N_z = 2048 \times 512 \times 1536$. The grid spacing is uniform in both the x (streamwise) and z (spanwise) directions. The resolution is $\Delta x^+ \times \Delta y^+ \times \Delta z^+ = 12 \times 6 \times 6$ at the channel center. The grid resolution at the wall is $\Delta y^+ = 0.017$. The reader is directed to [30] for further details of the DNS.

For training purposes, we use DNS data within the wall-normal distance range $10 < y^+$ and $y/\delta < 0.1$, where the flow behavior is universal (following the discussion in Sec. II A). Data in $y^+ < 10$ are precluded because no wall model would be needed when the small scales are also resolved. Data above $y/\delta = 0.1$ are not included for training because the flow away from the wall is usually resolved by LES grids.

We spatially filter the DNS data so that the filtered data conform with solutions of WMLES (see Fig. 6). A well-resolved LES will have data close to the wall available and a coarse-grid calculation will only have data slightly away from the wall available. To filter the DNS data to typical LES resolutions, we first define a virtual wall-adjacent LES grid. The grid is $\Delta x \times \Delta y \times \Delta z = \mathcal{R}h_{\mathrm{wm}} \times h_{\mathrm{wm}} \times \mathcal{R}h_{\mathrm{wm}}$ in size. Here $\mathcal{R} > 1$ is the aspect ratio of the first off-wall grid (see Fig. 6) and $10 < h^+_{\mathrm{wm}}$, $h_{\mathrm{wm}} < 0.1\delta$. If we conduct an LES using this grid, velocity information will only be available at a distance $y = h_{\mathrm{wm}}$ from the wall and at a wall-parallel resolution $\mathcal{R}h_{\mathrm{wm}} \times \mathcal{R}h_{\mathrm{wm}}$. The same is

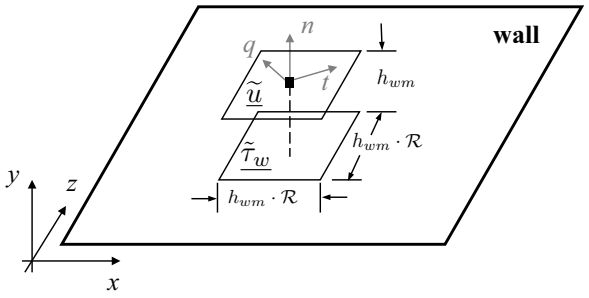

FIG. 6. Sketch of the spatial filtering of the DNS data. Both the instantaneous velocity and the instantaneous wall stress are filtered within a virtual LES computational cell of size $\Delta x \times \Delta y \times \Delta z = h_{\rm wm}\mathcal{R} \times h_{\rm wm} \times h_{\rm wm}\mathcal{R}$, where $\mathcal{R}$ varies from 1 to 12. A local coordinate system (t, q, n) is such that t is in the direction of the filtered velocity at $y = h_{\rm wm}$, n is the wall-normal direction, and q is perpendicular to both t and n.

true for the wall-shear stress; scales smaller than $\mathcal{R}h_{\rm wm} \times \mathcal{R}h_{\rm wm}$ will not be resolved by the LES grid. We use a top-hat filter in physical space and filter the DNS velocity at the distance $y = h_{\rm wm}$ in a square-shaped area of size $\mathcal{R}h_{\rm wm} \times \mathcal{R}h_{\rm wm}$. The same filtration is applied to the wall-shear stress. The filtered DNS data correspond to the LES solution within the wall-adjacent cell and will be used as our training data. Vectors t, n, and q define a local coordinate system, where t is the direction of the wall-parallel velocity, n is the wall-normal direction, and q is $t \times n$, with $\times$ denoting cross product here.

Since grid information is part of SGS models in LES, it is therefore quite natural to include grid information, i.e., $\mathcal{R}$, in data-based LES wall models. We vary $\mathcal{R}$ from 1 to 12, which are typical aspect ratio values of the near-wall computational cells in WMLES [77–87]. It is worth noting that a spatial filtering of the instantaneous DNS data will not be needed if the data are used for training a neural network for RANS simulation, because time averaging would have already removed all turbulence fluctuations. The filtered DNS data are fully three dimensional. We downsample evenly in both the streamwise and spanwise directions such that the training data contain about 2×10^9 input-output pairs from four independent realizations.

B. Neural networks

We train a few neural networks by successively including more flow information to the network input. A neural network models the expected wall stress given the model input, i.e., $\langle \tau_w | \text{input} \rangle$, where $\langle A|B \rangle$ is the expected value of A given condition B. In this section we compare the model predictions directly to DNS, i.e., an *a priori* study. The results of *a posteriori* studies will be presented in the next section, where the trained networks are used in WMLES. We note that the results of an *a priori* study are not at all related to the results of an *a posteriori* study for ML-based turbulence models [88–90]. Because the models are ultimately going to be tested in WMLES, we partition the training data into a training and a validation set, with no testing data. We stop training if the model error evaluated using the data in the validation set does not decrease for 15 epochs. Details of the neural networks are presented in Table I. Depending on the choice of inputs, the trained network may perform differently. For example, the outputs of NN1 will be independent of the grid aspect ratio and the outputs of NN3 will depend on local pressure gradients. The impact of our choices on the network inputs is shown in Figs. 7–9 by comparing the model results to the training data and in Figs. 14 and 15 by comparing the WMLES to DNS.

The first network uses solely the single-point wall-parallel velocity $u_\parallel$ at $h_{\rm wm}$. The model inputs are $|u_\parallel|/h_{\rm wm}$ and $\ln(h_{\rm wm}/y_0)/|u_\parallel|$, following the discussion in the preceding section. The local wall-shear stress is projected in the velocity direction and is used as the model output. The wall-shear stress in the q direction is neglected because $\langle \tau_{w,q} | u_\parallel, h_{\rm wm} \rangle = 0$. The neural net contains three

TABLE I. Details of the neural networks. Here NN denotes neural network and HL hidden layer. The tabulated hidden layer size contains the number of neurons within each hidden layer. A local coordinate system is set up such that the wall normal direction is n, the velocity direction is t, and the third direction is q (see Fig. 6 for details). In addition, $\nabla_q p$ denotes the pressure gradient in the q direction in the local coordinate system. All quantities are in wall units.

NN	HL size	Input	Output
NN1	(4,2,2)	$\frac{\lvert u_\parallel\rvert}{h_{\mathrm{wm}}}$, $\frac{\ln(h_{\mathrm{wm}}/y_0)}{\lvert u_\parallel\rvert}$	$\lvert\tau_{w,t}\rvert$
NN2	(6,4,3,3)	$\frac{\lvert u_\parallel\rvert}{h_{\mathrm{wm}}}$, $\frac{\ln(h_{\mathrm{wm}}/y_0)}{\lvert u_\parallel\rvert}$, $\mathcal{R}$	$\lvert\tau_{w,t}\rvert$
NN3	(8,8,6,4,4)	$\frac{\lvert u_\parallel\rvert}{h_{\mathrm{wm}}}$, $\frac{\ln(h_{\mathrm{wm}}/y_0)}{\lvert u_\parallel\rvert}$, $\mathcal{R}$, $\nabla_q p \frac{h_{\mathrm{wm}}}{\delta}$	$\lvert\tau_{w,t}\rvert$, $\lvert\tau_{w,q}\rvert$

hidden layers with four, two, and two neurons, respectively. For wall modeling, being cheap to evaluate is usually an equally important consideration as accuracy. A wall model is evaluated at every wall location and every time step in a three-dimensional scale-resolving simulation. Limiting the computation needed to evaluate the model (even if it is just a few multiplications and additions) is therefore critical to model performance. This is quite different from other commercial applications, e.g., image recognition, where evaluating a neural network is usually considered "free." The neural network architecture (in terms of hidden layers and neurons per layer) is determined through trial and error and is found to be a good trade-off between model accuracy and computational cost. Specifically, we start with a sufficiently large network and successively reduce the size of the network until we cannot get a reasonably accurate prediction in an *a priori* sense. Once the size of the neural network is determined, it takes about one day to train on a work station that contains 16 CPUs.

Figure 7 compares the model to the (filtered) DNS data at $y^+ \approx 10$ and $y^+ \approx 100$. The (filtered) DNS data and predictions of the zonal model [15] are included for comparison. In Fig. 7 we also show the PDF of the wall-parallel velocity $u_\parallel$. The DNS wall-shear stress is not solely a function of the near-wall velocity, leading to the observed data scattering. Compared to the zonal model, the neural network fits the data slightly better. Lacking the physical knowledge of flow at rare $u_\parallel$ values, the neural network levels off for $u_\parallel$ being large and small, even when the model error is weighted (following Sec. III A). Results at other wall-normal heights are similar and are not shown for brevity.

Next we include grid information, i.e., the grid aspect ratio $\mathcal{R}$, in the model input. The grid $\mathcal{R}$ is a measure of the LES filtering, i.e., a large $\mathcal{R}$ corresponds to a coarse grid in the wall-normal

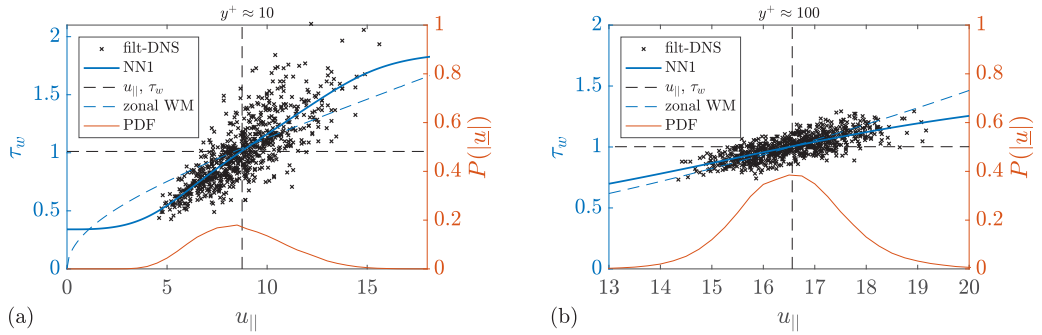

FIG. 7. (a) The left axis gives the wall-shear stress as a function of the wall-parallel velocity at $y^+ \approx 10$: ×, filtered DNS data; —, predictions of NN1; −−, the mean velocity and the mean wall-shear stress; −−, predictions of the zonal model. The right axis gives the PDF of the wall-parallel velocity: —, PDF of the filtered velocity. All quantities are normalized using wall units. (b) Same as (a) but at the wall-normal distance $y^+ \approx 100$.

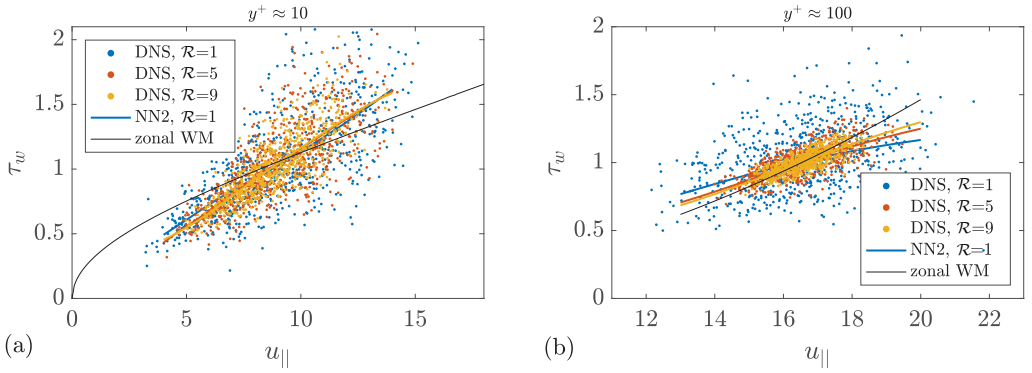

FIG. 8. (a) Wall shear stress as a function of the off-wall velocity at $y^+ \approx 10$. Different colors are used for data of different $\mathcal{R}$. The DNS data are dots and the neural network outputs are lines; WM denotes wall model. (The red and orange lines are neural network outputs for $\mathcal{R} = 5$ and 9, respectively.) The thin black line shows the prediction of the zonal model. (b) Same as (a) but at $y^+ \approx 100$.

directions, which is a large-sized LES filter. Including grid information is almost trivial for a data-based approach, despite the difficulty of interpreting results obtained from the trained network. For conventional modeling approaches, incorporating grid information needs a physical understanding of the coarse-grained near-wall turbulence [67,91], which is highly nontrivial and has barely been tried. From this perspective, physics-informed data-based approaches are more advantageous to conventional approaches.

The network contains four hidden layers, with six, four, four, and three neurons, respectively. We compare output from the trained neural network with the DNS data and the zonal model in Fig. 8. Direct numerical simulation shows less scattering for larger $\mathcal{R}$, due to the effects of filtering. The neural network output, on the other hand, is only a weak function of the grid aspect ratio $\mathcal{R}$.

Finally, we include pressure gradient in the crossflow direction q in the input. For NN1 and NN2, we have not modeled the wall-shear stress in the crossflow direction, which will be 0 by symmetry, i.e., $\langle \tau_{w,q} | u_{\parallel}, h_{\rm wm} \rangle = 0$. By including the pressure gradient in the crossflow direction, the symmetry breaks down and we can include wall-shear stress in the crossflow direction $\tau_{w,q}$ in model output. Following the discussion in Sec. II A, pressure gradient is included in the integral $\nabla_q p^+ h_{\rm wm}/\delta$. Figure 9 shows the stress in the crossflow direction as a function of the pressure gradient in the same direction. Here $\langle \tau_{w,q} | \nabla_q p^+ h_{\rm wm}/\delta \rangle \neq 0$ if $\nabla_q p^+ h_{\rm wm}/\delta \neq 0$. The scattering of DNS at $\nabla_q p^+ h_{\rm wm}/\delta = 0$

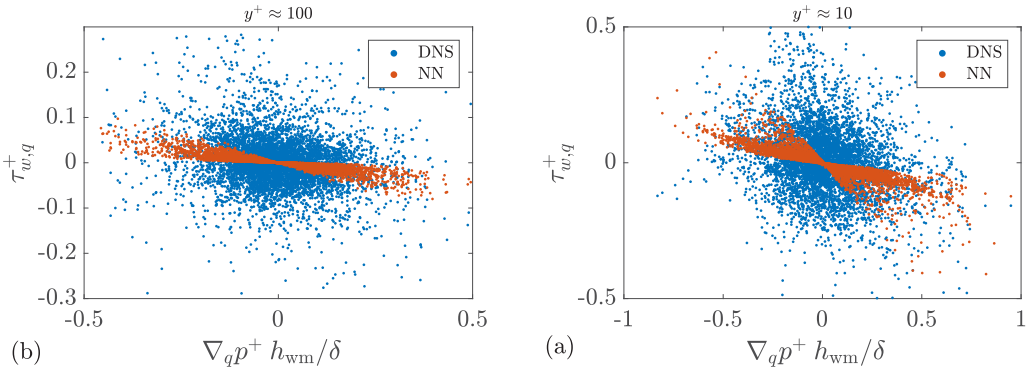

FIG. 9. (a) Crossflow wall-shear stress as a function of the crossflow pressure gradient for $h^+_{\rm wm} = 10$. (b) Same as (a) but for $h^+_{\rm wm} = 100$.

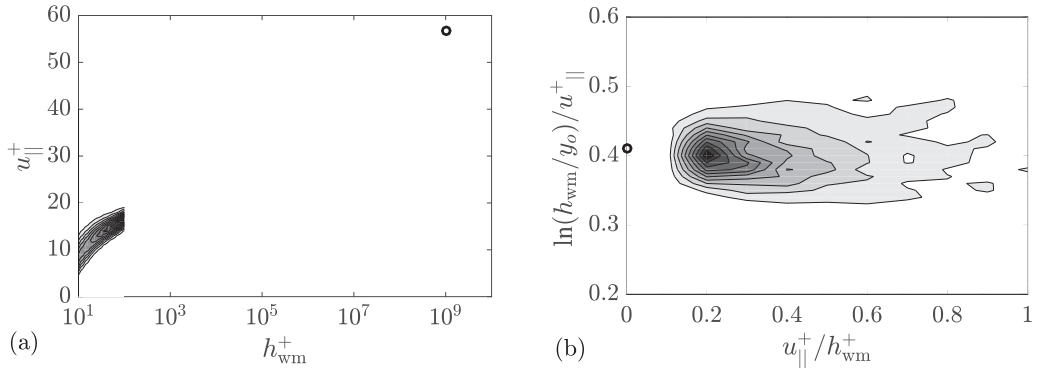

FIG. 10. (a) Parameter space in terms of h_{wm}^{+} and $u_{\parallel}^{+}$. The contour shows the PDF of the training data. The $\mathrm{Re}_\tau = 10^{10}$ channel will cover the parameter space around the circled position. (b) Same as (a) but for parameter space in terms of $\ln(h_{\mathrm{wm}}/y_0)/u_{\parallel}^{+}$ and $u_{\parallel}^{+}/h_{\mathrm{wm}}$.

shows that $\tau_{w,q}$ is not only a function of the pressure gradient in the crossflow direction. On the other hand, the scattering of the neural network outputs shows that $\tau_{w,q}$ also depends on the wall-parallel velocity and h_{wm}.

C. Wall-modeled LES

In this section we test the trained neural networks in WMLES. The neural networks are implemented and tested in the in-house code LESGO. LESGO has been used extensively for boundary-layer-flow calculations [92–97]. The code uses pseudospectral discretization in the streamwise and spanwise directions and a second-order finite difference in the wall-normal direction. A second-order Adam-Bashforth method is used for time stepping. The SGS stress is modeled using the scale-dependent Lagrangian Smagorinsky model [98–100]. Other details of the code may be found in Refs. [68,101].

Two test flows are considered: One is plane channel flow from $\mathrm{Re}_\tau = 10^3$ to $\mathrm{Re}_\tau = 10^{10}$ (which are at equal or higher Reynolds numbers than the training flow at $\mathrm{Re}_\tau = 10^3$); the other is a three-dimensional boundary layer at $\mathrm{Re}_\tau \approx 1000$, which involves strong nonequilibrium effects that are not present in the training data. For the channel flow case, the main challenge is the much higher Reynolds number of the test flows compared to the training flow, which may necessitate aggressive extrapolations and post challenges to data-based models. For three-dimensional boundary layers, the prime challenge is nonequilibrium effects.

To address the first challenge, we have regrouped the model inputs. Figure 10 shows the parameter space covered by the training data and approximately the parameter space of a $\mathrm{Re}_\tau = 10^{10}$ channel. The parameter space of a $\mathrm{Re}_\tau = 10^{10}$ channel is in the neighborhood of $h_{\mathrm{wm}}^{+} \approx 10^9$ and $u_{\parallel}^{+} = 1/\kappa \ln(h_{\mathrm{wm}}) + B = 56.8$. If the wall-parallel velocity $u_{\parallel}^{+}$ and the distance of the first off-wall grid h_{wm}^{+} are used directly for training, a neural network trained using data at $\mathrm{Re}_\tau = 1000$ will rely completely on extrapolation for predicting flow at $\mathrm{Re}_\tau \approx 10^{10}$ [Fig. 10(a)]. However, regrouping h_{wm}^{+} and $u_{\parallel}^{+}$ to $\ln(h_{\mathrm{wm}}/y_0)/u_{\parallel}^{+}$ and $u_{\parallel}^{+}/h_{\mathrm{wm}}^{+}$, a less aggressive extrapolation is needed for the $\mathrm{Re}_\tau = 10^{10}$ channel [Fig. 10(b)].

The second challenge is addressed, partly, as we are training using instantaneous realizations of the Navier-Stokes equation. While the channel is at equilibrium on average, the flow may be subject to nonequilibrium effects from time to time. Trained using instantaneous data that are subject to nonequilibrium effects, a network learns, albeit not directly, the effects of nonequilibrium terms on the wall-shear stress. The above discussion involves two aspects of wall modeling. One aspect is whether a wall model can transmit the resolved unsteadiness to the modeled wall-shear stress. The

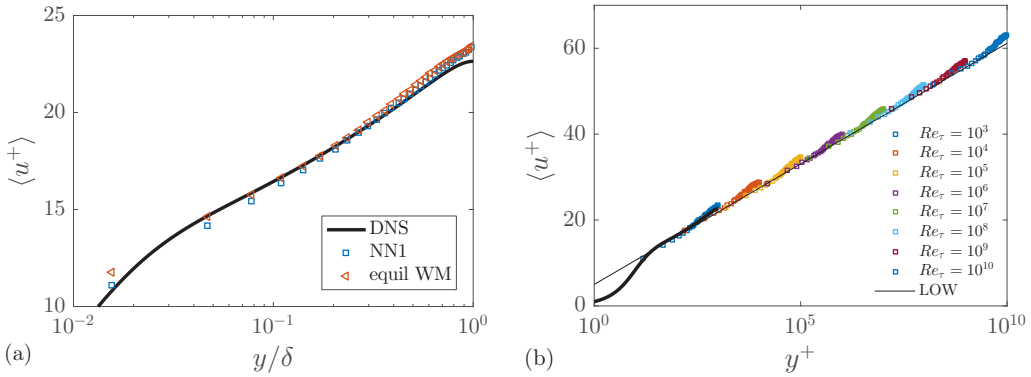

FIG. 11. (a) Mean velocity profiles as a function of wall-normal distance for a channel at $\mathrm{Re}_\tau = 1000$. Symbols show the results of WMLES. (b) Mean velocity profiles as a function of wall-normal distance for channel flow at friction Reynolds numbers from $\mathrm{Re}_\tau = 10^3$ to $\mathrm{Re}_\tau = 10^{10}$. The symbols show the results of WMLES. The thin solid line is the logarithmic law of the wall and the bold solid line is the mean profile of the DNS channel at $\mathrm{Re}_\tau = 1000$.

other aspect is whether a wall model can infer from the resolved LES some information about the unresolved near-wall flow and transfer that information to the modeled wall-shear stress. A simple equilibrium wall model can already transmit the resolved unsteadiness to the modeled wall-shear stress, but by imposing directly the law of the wall, the equilibrium wall model does not contain any information about the flow between the wall and the first off-wall grid. In a channel flow, any deviation from the law of the wall is a result of nonequilibrium effects and therefore velocity fluctuations carry information about nonequilibrium effects (although not directly). Hence trained using instantaneous data, a network "knows" the effects of the nonequilibrium terms, albeit in an implicit and indirect manner through the training data. This effect is probably clear from Fig. 7(b). Both the neural network and the equilibrium model transmit the resolved unsteadiness: The model predicted wall-shear stress is a function of the resolved near-wall velocity. However, the functional dependence of the wall-shear stress on the near-wall velocity does not follow the one suggested by the law of the wall. Instead, at velocity values that are away from the equilibrium value, the neural net predicts wall-shear stress that is slightly closer to the equilibrium value than the equilibrium wall model. Because the mean flow follows the law of the wall at $y^+ = 100$, the difference between the neural net and the equilibrium wall model is the result of the nonequilibrium terms that are nonzero instantaneously.

For canonical channel flow, WMLES is conducted for a half channel. A symmetry condition is imposed at the top boundary. A grid of size $N_x \times N_y \times N_z = 32 \times 32 \times 32$ is used for a computation domain of size $L_x \times L_y \times L_z = 2\pi\delta \times \delta \times 2\pi\delta$. The flow is driven by a constant pressure gradient. The flow statistics are obtained by averaging for ten flowthroughs, where $t_f = L_x/u_b$ is one flowthrough and u_b is the bulk velocity. The velocity at the first off-wall grid is filtered before being fed to the neural networks following Yang *et al.* [102]. Figure 11(a) shows the mean velocity profiles as functions of the wall-normal distance for the $\mathrm{Re}_\tau = 1000$ channel. Results of neural networks are shown only for NN1. For channel flow, WMLES with wall-shear stress modeled using NN2 and NN3 is the same as that from NN1 WMLES. The WMLESs using both NN1 and the algebraic equilibrium wall model follow the DNS profile closely. Figure 11(b) shows the mean profiles of NN1 WMLES at friction Reynolds numbers from $\mathrm{Re}_\tau = 10^3$ to 10^{10}. The neural network predicts the LOW, even though the network is not informed directly with this knowledge. Admittedly, using $\ln(h_{\mathrm{wm}}/y_0)$ as one of the inputs of the ML model may have helped the data-based model in finding the logarithmic LOW. Hence it is probably not surprising that the neural network trained using fully developed channel flow data at $\mathrm{Re}_\tau = 10^3$ can be used for flow at any Reynolds number. Figure 12

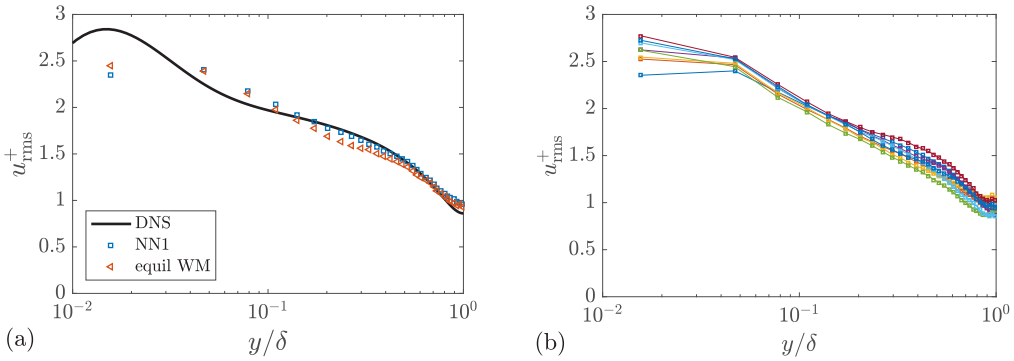

FIG. 12. Root mean square of the streamwise velocity for the same legend as in (a) Fig. 11(a) and (b) Fig. 11(b).

shows the root mean square of the streamwise velocity fluctuation. The difference between the network and the equilibrium wall model is not significant.

Next we consider a three-dimensional boundary-layer flow, where a mean spanwise pressure gradient $\partial p/\partial z = 10\rho u_\tau^2$ is impulsively imposed to a fully developed $\mathrm{Re}_\tau = 934$ channel flow, where u_τ is the friction velocity of the fully developed two-dimensional channel. The flow gradually develops a mean velocity in the spanwise direction, and until the flow reaches a statistically stationary state, the mean flow near the wall is not aligned with the wall-shear stress. We use WMLES for this transient flow. The computational domain is $L_x \times L_y \times L_z = 8\pi\delta \times \delta \times 4\pi\delta$, and a grid of size $N_x \times N_y \times N_z = 128 \times 32 \times 64$ is used. The boundary conditions are the same as the channel flow. Figures 13(a) and 13(b) show the contours of the instantaneous velocity

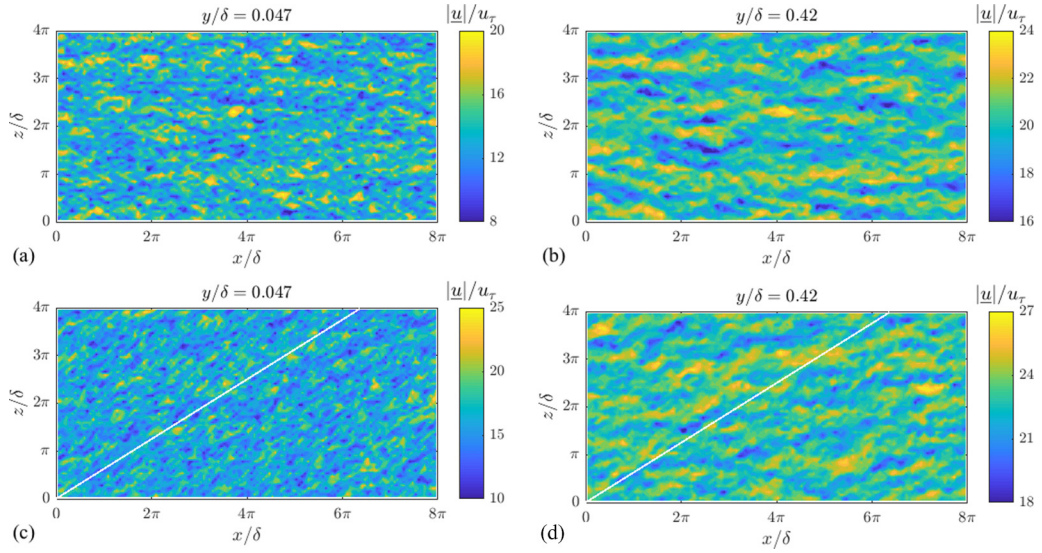

FIG. 13. (a) Contours of the instantaneous velocity at the wall-normal distance $y/\delta = 0.047$ at $t = 0$. (b) Same as (a) but at the wall-normal distance $y/\delta = 0.42$. (c) Same as (a) but at a time $t = \delta/u_\tau$. The solid white line is the direction of the shear stress at the wall. (d) Same as (c) but at a wall-normal distance $y/\delta = 0.42$. Here u_τ is the friction velocity of the two-dimensional channel. An underline denotes a vector and $|\cdot|$ is the norm of the bracketed quantity.

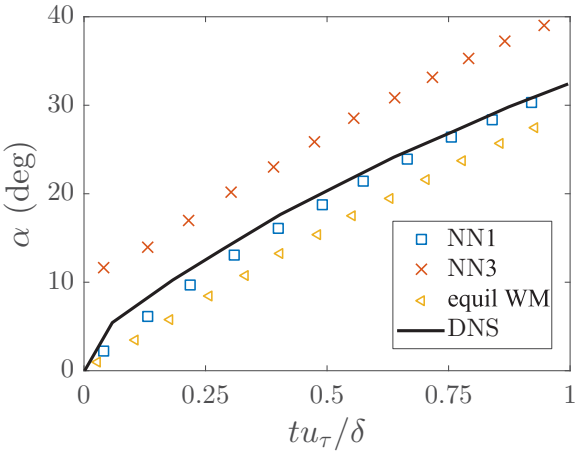

FIG. 14. Direction of the spatially averaged wall-shear stress as a function of the time. Here α is the angle between the spatially averaged wall-shear stress and the streamwise direction of the two-dimensional channel.

$\sqrt{u^2+v^2+w^2}$ at two wall-normal heights before the spanwise pressure gradient is imposed, which we take as $t=0$. Streaklike flow structures are found at both planes. The scales of the flow structures are smaller at $y=0.047\delta$ than at $y=0.42\delta$. Figures 13(c) and 13(d) show the contours of the instantaneous velocity at the same wall-normal heights at time $t=\delta/u_\tau$, where the streaks are aligned with an off-x direction. The white lines show the direction of the plane-averaged wall-shear stress. The near-wall streaks are aligned with the wall-shear stress already, but the streaks away from the wall are not. Comparing Figs. 13(a) and 13(c), there appear to be more structures in Fig. 13(c). Considering that the flow in Fig. 13(c) is at a higher Reynolds number than that in Fig. 13(a), it is probably expected that there is a larger number of flow structures in Fig. 13(c). In general, the flow in the outer region is well resolved by the LES grids. The equilibrium model and the neural network lead to qualitatively similar flow fields; we show only the results of NN1. Figure 14 shows the direction of the spatially averaged wall-shear stress as a function of time. NN1 follows the DNS measurements closely [103] and is found to be more accurate than the equilibrium wall model. The results of NN2 are very similar to NN1 and are not shown here for brevity. NN3, however, overpredicts the stress in the spanwise direction, although the pressure gradient was explicitly accounted for in the model. Figure 15 shows the spatially averaged spanwise velocity from $t=0$ to $tu_\tau/\delta=1.0$. The WMLES results using both the equilibrium model and the neural network follow the DNS quite closely. As the spanwise pressure is responsible for the momentum gain in the z direction, it is probably not surprising that the mean flow results are not very sensitive to near-wall turbulence modeling. In all, the data-based approach offers some advantages over the conventional equilibrium model in model accuracy. However, including more flow information does not necessarily lead to improved WMLES results. A detailed analysis of this flow and a comprehensive comparison between wall models were presented in Ref. [103], and the reader is directed to [103] for further details of this flow.

V. DISCUSSION

Data-based turbulence modeling faces very different challenges than conventional physics-based approaches. Among other challenges, the most pressing ones are (i) how to incorporate prior knowledge, e.g., a known scaling, in a neural network, without imposing that exact scaling; (ii) how to deal with sparse, incompletely sampled parameter space; (iii) whether a data-based model will be able to answer questions that are not directly registered around the training data; and (iv) how

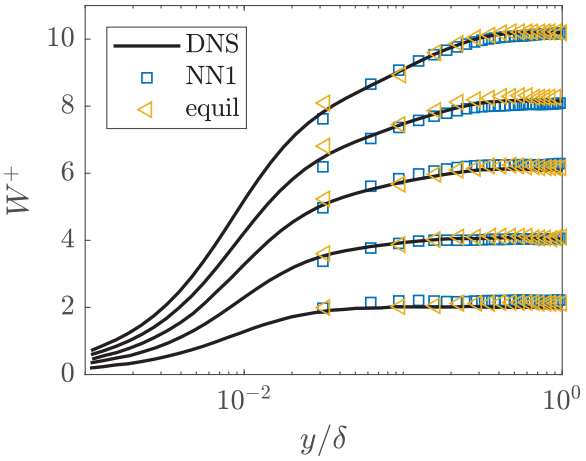

FIG. 15. Spatially averaged spanwise velocity as a function of the wall-normal distance at $tu_\tau/\delta \approx 0.2$, 0.4, 0.6, 0.8, and 1.0 (from bottom to top). The WMLES results are plotted every other point for a better presentation. Velocities are normalized by the friction velocity of the two-dimensional channel flow.

to train a model such that it can be used for predictive modeling. This work addresses the above challenges in the particular context of LES wall modeling, but issues including wall curvature, pressure gradients, and flow separation, remain unresolved. Considering of the success of machine learning in other disciplines, the general expectation is that, with the above challenges (i)–(iv) addressed, a data-based model will "learn" from the data and handle pressure gradient, curvature, flow separation, etc., when more data become available.

We show that, by incorporating prior knowledge on eddy population density, the trained network can be used as a predictive model in the context of LES wall modeling and answer questions that are not registered near the training data set. It is a consensus that to train a network for predictive causal modeling, one has to incorporate prior knowledge. The challenge has been how one can go about doing that. This work is devoted to answering the above question in the specific context of LES wall modeling. Nonetheless, the proposed methodology may well be used in a different context. Let us consider a variant of the example in Sec. III. Consider training a network to learn from some data the relation $y = x^2$, where x is the input and y is the output. The data are noisy, i.e., are $y = x^2(1 + r)$, where r is a random number that adds 10% noise to the data. The training data are between $x = 0$ and $x = 1$ and the objective is to predict y at $x = 10$. If we already know the scaling, we could directly use x^2 as network input, and by employing a rectified linear unit function as the activation function, the problem is readily solved. However, the network breaks if one employs, e.g., a sigmoid activation function. A more robust approach is to train the neural net for $y_2(x) = y(x)/x^2$, i.e., train using data pairs $(x^2, y/x^2)$. The trained network $y_2(x)$ can be used to compute y according to $y = y_2(x^2)x^2$. Doing that and training a network using data between 0 and 1, one may employ any activation function to predict y at $x = 10$ (results omitted for brevity). The proposition is as follows: Knowledge of the functional dependence of y on x, if known, should be used such that, for network training, the outputs are normalized by the known scaling. A trivial question may be, if we already know the scaling, why do we still need machine learning? The answer is that known scalings are often first-order approximations of the reality and we can use machine learning for second-order effects.

VI. CONCLUSION

The usefulness of machine learning techniques is examined in the context of LES wall modeling. Limited by the availability of DNS data, in the foreseeable future, learning with the objective of wall

modeling will have to rely on data at moderate Reynolds numbers and flows in simple geometries. In this work we trained neural networks using fully developed channel flow data at $\mathrm{Re}_\tau = 1000$ and the extrapolation ability of the trained networks was carefully tested. The trained networks NN1, NN2, and NN3 captured the law of the wall at the Reynolds number of the training data and at other Reynolds numbers, although the network was not directly informed with this knowledge. In addition, network NN1 outperformed the conventional equilibrium wall model in a nonequilibrium flow, although the training data for this network was obtained from the DNS of fully developed equilibrium channel flow. The outcome of the work is a trained neural network that could be used directly in WMLES of incompressible turbulent flows.

It is crucial to note that this work is but a first step towards data-based WMLES. In this work we incorporated our knowledge of the mean flow scalings in network training. The advantage of a data-based approach is its ability to go beyond a known scaling by learning from data. Since we also know how the mean flow behaves when subject to weak nonequilibrium effects [16], we may incorporate that knowledge in a data-based model when training a neural net using DNS data of flows other than channel flow. For example, in order to train a neural net to handle pressure gradient, we could use DNS of Couette flow with a spanwise pressure gradient; in order to train a neural net to handle stream curvature, we could use DNS of flow between two rotating cylinders, etc. However, because DNSs of such flows are not yet publicly accessible, we will leave pressure gradients, curvature, and other nonequilibrium effects to future investigations. Many challenges remain to be addressed. First, increasing the size of the training data set, e.g., including DNS of flows with separation, could probably further improve model accuracy, but this exercise is only possible if more DNS data sets including those in Refs. [104–106] are made publicly available. Second, it is not entirely clear how much flow information needs to be included for model input. For instance, we have shown with a high degree of confidence that including the pressure gradient in the crossflow direction does not aid model accuracy improvements. Third, improving accuracy while not letting computational costs skyrocket usually implies a trade-off in terms of the number of hidden layers in a neural network, since training deep networks (deep learning) can be prohibitive in terms of hours needed. This exact balance is yet to be found. Finally, other ML techniques such as random forests and convolutional neural networks (CNNs) may prove to be more useful for WMLES than fully connected neural networks; CNNs in particular are a possibility which needs to be explored in the future.

The trained networks are publicly available through the FCPRL website [107], and can be shared through email.

ACKNOWLEDGMENT

X.I.A.Y. would like to thank Penn State University for financial support. The computations are performed on ACI at Penn State and XSEDE.

[1] J. Jiménez, Turbulent flows over rough walls, Annu. Rev. Fluid Mech. **36**, 173 (2004).
[2] I. Marusic, B. McKeon, P. Monkewitz, H. Nagib, A. Smits, and K. Sreenivasan, Wall-bounded turbulent flows at high Reynolds numbers: Recent advances and key issues, Phys. Fluids **22**, 065103 (2010).
[3] A. J. Smits, B. J. McKeon, and I. Marusic, High Reynolds number wall turbulence, Annu. Rev. Fluid Mech. **43**, 353 (2011).
[4] P. Moin and K. Mahesh, Direct numerical simulation: A tool in turbulence research, Annu. Rev. Fluid Mech. **30**, 539 (1998).
[5] H. Choi and P. Moin, Grid-point requirements for large eddy simulation: Chapman's estimates revisited, Phys. Fluids **24**, 011702 (2012).
[6] D. R. Chapman, Computational aerodynamics development and outlook, AIAA J. **17**, 1293 (1979).

[7] U. Piomelli and E. Balaras, Wall-layer models for large-eddy simulations, Annu. Rev. Fluid Mech. **34**, 349 (2002).
[8] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis, CFD vision 2030 study: A path to revolutionary computational aerosciences, NASA Langley Research Center Report No. NASA/CR-2014-218178, 2014 (unpublished), available at http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20140003093.pdf
[9] P. Moin, J. Bodart, S. Bose, and G. I. Park, Wall-modeling in complex turbulent flows, in *Advances in Fluid-Structure Interaction*, edited by M. Braza, A. Bottaro, and M. Thompson (Springer, Berlin, 2016), pp. 207–219.
[10] J. Larsson, S. Kawai, J. Bodart, and I. Bermejo-Moreno, Large eddy simulation with modeled wall-stress: Recent progress and future directions, Mech. Eng. Rev. **3**, 15 (2016).
[11] S. T. Bose and G. I. Park, Wall-modeled large-eddy simulation for complex turbulent flows, Annu. Rev. Fluid Mech. **50**, 535 (2018).
[12] U. Schumann, Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli, J. Comput. Phys. **18**, 376 (1975).
[13] G. I. Park and P. Moin, An improved dynamic non-equilibrium wall-model for large eddy simulation, Phys. Fluids **26**, 37 (2014).
[14] E. Balaras and C. Benocci, *Subgrid-Scale Models in Finite-Difference Simulations of Complex Wall Bounded Flows* (AGARD, Neuilly-Sur-Seine, 1994).
[15] S. Kawai and J. Larsson, Wall-modeling in large eddy simulation: Length scales, grid resolution, and accuracy, Phys. Fluids **24**, 015105 (2012).
[16] X. I. A. Yang, J. Sadique, R. Mittal, and C. Meneveau, Integral wall model for large eddy simulations of wall-bounded turbulent flows, Phys. Fluids **27**, 025112 (2015).
[17] D. Chung and D. Pullin, Large-eddy simulation and wall modelling of turbulent channel flow, J. Fluid Mech. **631**, 281 (2009).
[18] M. Inoue, R. Mathis, I. Marusic, and D. Pullin, Inner-layer intensities for the flat-plate turbulent boundary layer combining a predictive wall-model with large-eddy simulations, Phys. Fluids **24**, 075102 (2012).
[19] X. I. A. Yang and M. F. Howland, Implication of Taylor's hypothesis on measuring flow modulation, J. Fluid Mech. **836**, 222 (2018).
[20] S. Bose and P. Moin, A dynamic slip boundary condition for wall-modeled large-eddy simulation, Phys. Fluids **26**, 015104 (2014).
[21] H. J. Bae, A. Lozano-Duran, S. Bose, and P. Moin, Turbulence intensities in large-eddy simulation of wall-bounded flows, Phys. Rev. Fluids **3**, 014610 (2018).
[22] W. Cheng, D. Pullin, R. Samtaney, W. Zhang, and W. Gao, Large-eddy simulation of flow over a cylinder with Re_d from 3.9×10^3 to 8.5×10^5: A skin-friction perspective, J. Fluid Mech. **820**, 121 (2017).
[23] I. Bermejo-Moreno, L. Campo, J. Larsson, J. Bodart, D. Helmer, and J. K. Eaton, Confinement effects in shock wave/turbulent boundary layer interactions through wall-modelled large-eddy simulations, J. Fluid Mech. **758**, 5 (2014).
[24] J. Larsson, S. Laurence, I. Bermejo-Moreno, J. Bodart, S. Karl, and R. Vicquelin, Incipient thermal choking and stable shock-train formation in the heat-release region of a scramjet combustor. Part II: Large eddy simulations, Combust. Flame **162**, 907 (2015).
[25] P. S. Iyer, G. I. Park, and M. R. Malik, Application of wall-modeled LES to turbulent separated flows, *Proceedings of the 69th Annual Meeting of the APS Division of Fluid Dynamics, Portland, 2016* (APS, Ridge, 2016).
[26] X. I. A. Yang, J. Sadique, R. Mittal, and C. Meneveau, Exponential roughness layer and analytical model for turbulent boundary layer flow over rectangular-prism roughness elements, J. Fluid Mech. **789**, 127 (2016).
[27] X. I. A. Yang, J. Urzay, S. Bose, and P. Moin, Aerodynamic heating in wall-modeled large-eddy simulation of high-speed flows, AIAA J. **56**, 731 (2017).
[28] X. Yang, J. Urzay, and P. Moin, *Heat-Transfer Rates in Equilibrium-Wall-Modeled LES of Supersonic Turbulent Flows*, Center for Turbulence Research, Annual Research Briefs (Stanford University, Stanford, 2016), pp. 3–15.

[29] R. D. Moser, J. Kim, and N. N. Mansour, Direct numerical simulation of turbulent channel flow up to $\mathrm{Re}_\tau = 590$, Phys. Fluids **11**, 943 (1999).
[30] J. Graham, K. Kanov, X. I. A. Yang, M. Lee, N. Malaya, C. C. Lalescu, R. Burns, G. Eyink, A. Szalay, R. D. Moser, and C. Meneveau, A Web services accessible database of turbulent channel flow and its use for testing a new integral wall model for LES, J. Turbul. **17**, 181 (2016).
[31] M. Lee and R. D. Moser, Direct numerical simulation of turbulent channel flow up to $\mathrm{Re}_\tau \approx 5200$, J. Fluid Mech. **774**, 395 (2015).
[32] S. Hoyas and J. Jiménez, Scaling of the velocity fluctuations in turbulent channels up to $\mathrm{Re}_\tau = 2003$, Phys. Fluids **18**, 011702 (2006).
[33] T. B. Hedley and J. F. Keffer, Turbulent/non-turbulent decisions in an intermittent flow, J. Fluid Mech. **64**, 625 (1974).
[34] Y. Yamamoto and Y. Tsuji, Numerical evidence of logarithmic regions in channel flow at $\mathrm{Re}_\tau = 8000$, Phys. Rev. Fluids **3**, 012602 (2018).
[35] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE Computer Society, Washington, DC, 2014), pp. 1725–1732.
[36] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal Process. Mag. **29**, 82 (2012).
[37] B. Tracey, K. Duraisamy, and J. Alonso, Application of supervised learning to quantify uncertainties in turbulence and combustion modeling, *Proceedings of the 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Dallas/Ft. Worth Region, 2013* (AIAA, Reston, 2013), paper 2013-0259.
[38] B. D. Tracey, K. Duraisamy, and J. J. Alonso, *Proceedings of the 53rd AIAA Aerospace Sciences Meeting* (AIAA, Reston, 2015), paper 2015-1287.
[39] K. Duraisamy, Z. J. Zhang, and A. P. Singh, *Proceedings of the 53rd AIAA Aerospace Sciences Meeting* (Ref. [38]), paper 2015-1284.
[40] J. Ling, R. Jones, and J. Templeton, Machine learning strategies for systems with invariance properties, J. Comput. Phys. **318**, 22 (2016).
[41] J. Ling, A. Kurzawski, and J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, J. Fluid Mech. **807**, 155 (2016).
[42] J.-L. Wu, J.-X. Wang, and H. Xiao, A Bayesian calibration-prediction method for reducing model-form uncertainties with application in RANS simulations, Flow Turbul. Combust. **97**, 761 (2016).
[43] J.-X. Wang, J.-L. Wu, and H. Xiao, Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data, Phys. Rev. Fluids **2**, 034603 (2017).
[44] J.-L. Wu, H. Xiao, and E. G. Paterson, Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework, Phys. Rev. Fluids **3**, 074602 (2018).
[45] K. Duraisamy, G. Iaccarino, and H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid Mech. **51**, 357 (2019).
[46] R. N. King, P. E. Hamlington, and W. J. A. Dahm, Autonomic closure for turbulence simulations, Phys. Rev. E **93**, 031301 (2016).
[47] M. Gamahara and Y. Hattori, Searching for turbulence models by artificial neural network, Phys. Rev. Fluids **2**, 054604 (2017).
[48] R. Maulik and O. San, A neural network approach for the blind deconvolution of turbulent flows, J. Fluid Mech. **831**, 151 (2017).
[49] A. Vollant, G. Balarac, and C. Corre, Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures, J. Turbul. **18**, 854 (2017).
[50] Z. J. Zhang and K. Duraisamy, Machine learning methods for data-driven turbulence modeling, *Proceedings of the 22nd AIAA Computational Fluid Dynamics Conference* (AIAA, Reston, 2015), paper 2015-2460.
[51] M. Milano and P. Koumoutsakos, Neural network modeling for near wall turbulent flow, J. Comput. Phys. **182**, 1 (2002).

[52] M. Ma, J. Lu, and G. Tryggvason, Using statistical learning to close two-fluid multiphase flow equations for a simple bubbly system, Phys. Fluids **27**, 092101 (2015).
[53] M. Ma, J. Lu, and G. Tryggvason, Using statistical learning to close two-fluid multiphase flow equations for bubbly flows in vertical channels, Int. J. Multiphase Flow **85**, 336 (2016).
[54] F. R. Menter, Review of the shear-stress transport turbulence model experience from an industrial perspective, Int. J. Comput. Fluid D **23**, 305 (2009).
[55] C. Meneveau and J. Katz, Scale-invariance and turbulence models for large-eddy simulation, Annu. Rev. Fluid Mech. **32**, 1 (2000).
[56] K. Lilly, The representation of small-scale turbulence in numerical simulation experiments, NCAR Report No. 123, 1966 (unpublished).
[57] ERCOFTAC, ERCOFTAC classic collection database, available at http://cfd.mace.manchester.ac.uk/ercoftac/
[58] C. L. Rumsey, NASA Langley Research Center, Turbulence modeling resource, 2017, https://turbmodels.larc.nasa.gov
[59] AGARD, A selection of test cases for the validation of large-eddy simulations of turbulent flows, AGARD Report No. 345, available at http://torroja.dmt.upm.es/turbdata/agard/
[60] Y. Li, E. Perlman, M. Wan, Y. Yang, C. Meneveau, R. Burns, S. Chen, A. Szalay, and G. Eyink, A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence, J. Turbul. **9**, N31 (2008).
[61] Johns Hopkins turbulence databases, available at http://turbulence.pha.jhu.edu
[62] R. S. Michalski, Understanding the nature of learning: Issues and research directions, Mach. Learn. **2**, 3 (1986).
[63] G. Piatetsky-Shapiro, Knowledge discovery in real databases: A report on the IJCAI-89 workshop, AI Mag. **11**, 68 (1990).
[64] J. Kim, P. Moin, and R. Moser, Turbulence statistics in fully developed channel flow at low Reynolds number, J. Fluid Mech. **177**, 133 (1987).
[65] S. B. Pope, *Turbulent Flows* (Cambridge University Press, Cambridge, 2001).
[66] I. Marusic, J. P. Monty, M. Hultmark, and A. J. Smits, On the logarithmic region in wall turbulence, J. Fluid Mech. **716**, R3 (2013).
[67] X. Yang, S. Bose, and P. Moin, *A Physics-Based Interpretation of the Slip-Wall LES Model*, Center for Turbulence Research, Annual Research Briefs (Stanford University, Stanford, 2017), pp. 65–74.
[68] X. I. A. Yang and M. Abkar, A hierarchical random additive model for passive scalars in wall-bounded flows at high Reynolds numbers, J. Fluid Mech. **842**, 354 (2018).
[69] X. I. A. Yang and C. Meneveau, Hierarchical random additive model for wall-bounded flows at high Reynolds numbers, Fluid Dyn. Res. **51**, 011405 (2019).
[70] X. I. A. Yang, I. Marusic, and C. Meneveau, Hierarchical random additive process and logarithmic scaling of generalized high order, two-point correlations in turbulent boundary layer flow, Phys. Rev. Fluids **1**, 024402 (2016).
[71] D. Krug, X. I. A. Yang, C. M. De Silva, R. Ostilla-Mónico, R. Verzicco, I. Marusic, and D. Lohse, Statistics of turbulence in the energy-containing range of Taylor-Couette compared to canonical wall-bounded flows, J. Fluid Mech. **830**, 797 (2017).
[72] X. I. A. Yang, C. Meneveau, I. Marusic, and L. Biferale, Extended self-similarity in moment-generating-functions in wall-bounded turbulence at high Reynolds number, Phys. Rev. Fluids **1**, 044405 (2016).
[73] J. Ling and J. Templeton, Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty, Phys. Fluids **27**, 085103 (2015).
[74] L. Bottou, Large-scale machine learning with stochastic gradient descent, in *Proceedings of COMPSTAT'2010*, edited by Y. Lechevallier and G. Saporta (Springer, Berlin, 2010), pp. 177–186.
[75] K. Levenberg, A method for the solution of certain non-linear problems in least squares, Q. Appl. Math. **2**, 164 (1944).
[76] S. I. Gallant, *Neural Network Learning and Expert Systems* (MIT Press, Cambridge, 1993).
[77] X. I. A. Yang, On the mean flow behaviour in the presence of regional-scale surface roughness heterogeneity, Bound.-Layer Meteorol. **161**, 127 (2016).

[78] X. I. A. Yang and C. Meneveau, Large eddy simulations and parameterisation of roughness element orientation and flow direction effects in rough wall boundary layers, J. Turbul. **17**, 1072 (2016).
[79] F. Nicoud, J. Baggett, P. Moin, and W. Cabot, Large eddy simulation wall-modeling based on suboptimal control theory and linear stochastic estimation, Phys. Fluids **13**, 2968 (2001).
[80] J. A. Templeton, M. Wang, and P. Moin, An efficient wall model for large-eddy simulation based on optimal control theory, Phys. Fluids **18**, 025101 (2006).
[81] R. J. Stevens, M. Wilczek, and C. Meneveau, Large-eddy simulation study of the logarithmic law for second-and higher-order moments in turbulent wall-bounded flow, J. Fluid Mech. **757**, 888 (2014).
[82] W. Anderson and C. Meneveau, Dynamic roughness model for large-eddy simulation of turbulent flow over multiscale, fractal-like rough surfaces, J. Fluid Mech. **679**, 288 (2011).
[83] W. Anderson and C. Meneveau, A large-eddy simulation model for boundary-layer flow over surfaces with horizontally resolved but vertically unresolved roughness elements, Bound.-Layer Meteorol. **137**, 397 (2010).
[84] W. Anderson, J. M. Barros, K. T. Christensen, and A. Awasthi, Numerical and experimental study of mechanisms responsible for turbulent secondary flows in boundary layer flows over spanwise heterogeneous roughness, J. Fluid Mech. **768**, 316 (2015).
[85] D. Willingham, W. Anderson, K. T. Christensen, and J. M. Barros, Turbulent boundary layer flow over transverse aerodynamic roughness transitions: Induced mixing and flow characterization, Phys. Fluids **26**, 025111 (2014).
[86] X. Zhu, G. V. Iungo, S. Leonardi, and W. Anderson, Parametric study of urban-like topographic statistical moments relevant to a priori modelling of bulk aerodynamic parameters, Bound.-Layer Meteorol. **162**, 231 (2017).
[87] J. Yang and W. Anderson, Numerical study of turbulent channel flow over surfaces with variable spanwise heterogeneities: Topographically-driven secondary flows affect outer-layer similarity of turbulent length scales, Flow Turbul. Combust. **100**, 1 (2018).
[88] S. V. Poroseva, J. D. Colmenares F., and S. M. Murman, On the accuracy of RANS simulations with DNS data, Phys. Fluids **28**, 115102 (2016).
[89] R. L. Thompson, L. E. B. Sampaio, F. A. V. de Bragança Alves, L. Thais, and G. Mompean, A methodology to evaluate statistical errors in DNS data of plane channel flows, Comput. Fluids **130**, 1 (2016).
[90] J. Wu, H. Xiao, R. Sun, and Q. Wang, RANS equations with Reynolds stress closure can be ill-conditioned, arXiv:1803.05581.
[91] X. I. A. Yang and A. Lozano-Durán, A multifractal model for the momentum transfer process in wall-bounded flows, J. Fluid Mech. **824**, R2 (2017).
[92] M. Abkar and F. Porté-Agel, A new boundary condition for large-eddy simulation of boundary-layer flow over surface roughness transitions, J. Turbul. **13**, N23 (2012).
[93] M. Abkar and F. Porté-Agel, Mean and turbulent kinetic energy budgets inside and above very large wind farms under conventionally neutral condition, Renew. Energ. **70**, 142 (2014).
[94] M. Abkar and F. Porté-Agel, A new wind-farm parameterization for large-scale atmospheric models, J. Renew. Sustain. Energ. **7**, 013121 (2015).
[95] M. Abkar, H. J. Bae, and P. Moin, Minimum-dissipation scalar transport model for large-eddy simulation of turbulent flows, Phys. Rev. Fluids **1**, 041701 (2016).
[96] A. Andren, A. R. Brown, P. J. Mason, J. Graf, U. Schumann, C.-H. Moeng, and F. T. Nieuwstadt, Large-eddy simulation of a neutrally stratified boundary layer: A comparison of four computer codes, Q. J. R. Meteorol. Soc. **120**, 1457 (1994).
[97] J. D. Albertson and M. B. Parlange, Surface length-scales and shear stress: Implications for land-atmosphere interaction over complex terrain, Water Resour. Res. **35**, 2121 (1999).
[98] F. Porté-Agel, C. Meneveau, and M. B. Parlange, A scale-dependent dynamic model for large-eddy simulation: Application to a neutral atmospheric boundary layer, J. Fluid Mech. **415**, 261 (2000).
[99] E. Bou-Zeid, C. Meneveau, and M. Parlange, A scale-dependent lagrangian dynamic model for large eddy simulation of complex turbulent flows, Phys. Fluids **17**, 025105 (2005).

[100] R. Stoll and F. Porté-Agel, Dynamic subgrid-scale models for momentum and scalar fluxes in large-eddy simulations of neutrally stratified atmospheric boundary layers over heterogeneous terrain, Water Resour. Res. **42**, W01409 (2006).

[101] F. Porté-Agel, A scale-dependent dynamic model for scalar transport in large-eddy simulations of the atmospheric boundary layer, Bound.-Layer Meteorol **112**, 81 (2004).

[102] X. I. A. Yang, G. I. Park, and P. Moin, Log-layer mismatch and modeling of the fluctuating wall stress in wall-modeled large-eddy simulations, Phys. Rev. Fluids **2**, 104601 (2017).

[103] M. Giometto, A. Lozano-Duran, G. Park, and P. Moin, *Three-Dimensional Transient Channel Flow at Moderate Reynolds Numbers: Analysis and Wall Modeling*, Center for Turbulence Research, Annual Research Briefs (Stanford University, Stanford, 2017), pp. 65–74.

[104] S. Leonardi and I. P. Castro, Channel flow over large cube roughness: A direct numerical simulation study, J. Fluid Mech. **651**, 519 (2010).

[105] N. Sandham, E. Schülein, A. Wagner, S. Willems, and J. Steelant, Transitional shock-wave/boundary-layer interactions in hypersonic flow, J. Fluid Mech. **752**, 349 (2014).

[106] H. Abe, Reynolds-number dependence of wall-pressure fluctuations in a pressure-induced turbulent separation bubble, J. Fluid Mech. **833**, 563 (2017).

[107] https://sites.psu.edu/xiangyangmne/