

Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework

Jin-Long Wu,^{*} Heng Xiao,[†] and Eric Paterson

Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, Virginia 24060, USA



(Received 8 January 2018; published 10 July 2018)

Reynolds-averaged Navier-Stokes (RANS) equations are widely used in engineering turbulent flow simulations. However, RANS predictions may have large discrepancies due to the uncertainties in modeled Reynolds stresses. Recently, Wang *et al.* demonstrated that machine learning can be used to improve the RANS modeled Reynolds stresses by leveraging data from high-fidelity simulations [J.-X. Wang *et al.*, *Phys. Rev. Fluids* **2**, 034603 (2017)]. However, solving for mean flows from the improved Reynolds stresses still poses significant challenges due to potential ill-conditioning of RANS equations with Reynolds stress closures. Enabling improved predictions of mean velocities is of profound practical importance, because often the velocity and its derived quantities (quantities of interest, e.g., drag, lift, and surface friction), and not the Reynolds stress itself, are of ultimate interest in RANS simulations. To this end, we present a comprehensive framework for augmenting turbulence models with physics-informed machine learning, illustrating a complete workflow from identification of input features to final prediction of mean velocities. This work has two innovations. First, we demonstrate a systematic procedure to generate mean flow features based on the integrity basis for mean flow tensors. Second, we propose using machine learning to predict linear and nonlinear parts of the Reynolds stress tensor separately. Inspired by the finite polynomial representation of tensors in classical turbulence modeling, such a decomposition is instrumental in overcoming the ill-conditioning of RANS equations. Numerical tests demonstrate the merits of the proposed framework.

DOI: [10.1103/PhysRevFluids.3.074602](https://doi.org/10.1103/PhysRevFluids.3.074602)

I. INTRODUCTION

Numerical simulations based on Reynolds-averaged Navier-Stokes (RANS) models are still the workhorse tool in engineering design involving turbulent flows. However, predictions from RANS simulations are known to have large discrepancies in many flows of engineering relevance, including those with swirl, pressure gradients, or mean streamline curvature [1]. It is a consensus that the dominant cause for such discrepancies is the RANS-modeled Reynolds stresses [2]. In light of the long stagnation in traditional turbulence modeling, researchers [3–7] explored machine learning as an alternative to improve RANS modeling by leveraging data from high-fidelity simulations.

A. Data-driven methods for reducing model discrepancies in RANS simulations

Data-driven methods have been devised to calibrate the model form uncertainties in RANS simulations based on optimization [8] and Bayesian inference approaches [8–12]. However, these data-driven calibration approaches inferred the model discrepancies of a given flow and thus lack the generalization capabilities for predicting flows with vastly different geometries from the calibration

^{*}Email: jinlong@vt.edu

[†]Corresponding author. Email: hengxiao@vt.edu

flow. Therefore, follow-on works of these researchers build data-driven turbulence models in the mean flow feature space (as opposed to physical space). Such an approach enables prediction for flows in different geometries yet with similar physics (e.g., curved pipes and wing-body juncture, both featuring secondary flows driven by Reynolds stress anisotropies). Duraisamy and co-workers [3,13,14] used nondimensional flow variables as the input features and a multiplicative correction term in the Spalart-Allmaras model as the machine learning output. Their machine-learning-augmented model demonstrated good generalization capabilities within a class of flows around airfoils [14]. Ling *et al.* [5] pointed out the importance of embedding the tensorial invariance properties in the machine learning process and used this approach to predict the Reynolds stress with a deep neural network [6]. Wang *et al.* [7] built a machine learning model to predict the discrepancies in the RANS modeled Reynolds stresses. Encouraging results have been demonstrated in prediction of Reynolds stresses in two sets of canonical flows (separated flows over periodic hills and secondary flows in a square duct). However, the machine-learning-predicted Reynolds stress leads to large error of solved mean velocities when substituted into the RANS equations. Such an ill-conditioning issue is a common challenge for data-driven Reynolds stress models that must be addressed to unleash the power of such models.

A distinctly different approach of data-driven modeling was pursued by Weatheritt and Sandberg [15,16], who used symbolic regression and gene expression programming to develop algebraic Reynolds stress models. To some extent, their approach is a combination of traditional modeling and data-driven modeling methods reviewed above. Specifically, while data-driven methods are used to obtain their model, the end product is an algebraic Reynolds model in the traditional sense. As such, the ill-conditioning issue for their model would be similar to the traditional models with explicit analytical forms and not the data-driven models.

B. Conditioning of data-driven Reynolds stress models

In this work we refer to solving the RANS equations for mean velocities with a given Reynolds stress field as propagation, which is a critical component in data-driven turbulence modeling. Poroseva *et al.* [17] referred to such simulation as RANS-DNS simulations. Admittedly, both terms could cause potential confusion and thus warrant the explicit clarification here.

Recently, several researchers have observed that small errors in the Reynolds stresses can be amplified to large errors in the mean velocities when solving the RANS equations with specified Reynolds stresses. Thompson *et al.* [18] propagated Reynolds stresses in channel flows at a wide range of frictional Reynolds numbers ($Re_\tau = 180\text{--}5200$) to mean velocities by using several reputed direct numerical simulation (DNS) databases. They reported that the propagated mean velocities can deviate significantly from the mean velocities from the DNS, especially for flows at high Reynolds numbers (notably $Re_\tau = 5200$). Poroseva *et al.* [17] also made similar observations and Poroseva [19] further pointed out that the discrepancies between the propagated velocities and DNS velocities were observed for flows at Reynolds numbers as low as $Re_\tau = 395$, depending on the data set used. Considering that errors in DNS Reynolds stress are typically less than 0.5% [20,21], these exercises of propagating DNS Reynolds stresses to mean velocities thus represent an ideal scenario for data-driven turbulence models with negligible modeling errors. Wu *et al.* [21] explained such observations by pointing out that RANS equations with Reynolds stress closure models can be ill-conditioned. They further proposed a condition number function defined based on the local velocities to quantify the ill-conditioning. For plane channel flows, the local condition number does increase with Reynolds number, which thus explained the increased ill-conditioning with increasing Reynolds number. In contrast, the traditional, matrix-based conditional number was not able to explain such observations. A physical explanation is that viscous stresses are negligible at high Reynolds numbers and the mean velocity is determined by the dependence of the Reynolds stresses on the mean velocity gradients. Therefore, obtaining the mean velocity would fail from an *a priori* specification of the Reynolds stresses. A unique issue associated with data-driven modeling is that it can be difficult or

even impossible to treat the Reynolds stress implicitly as in traditional models, and in such cases segregated solvers are the only option. A more detailed discussion can be found in [21].

A similar ill-conditioning issue also exists in traditional Reynolds stress models (RSMs), where Reynolds stresses are obtained by solving transport equations. To enhance the stabilities, Jarklic and co-workers [22,23] blended τ_{RSM} obtained from solving the Reynolds stress transport equation with that given by a linear eddy viscosity model, i.e., $\tau = \alpha\tau_{\text{RSM}} + (1 - \alpha)\tau_{\text{LEM}}$. However, the specification of a blending factor α is largely *ad hoc* and lacks physical basis. In this work we aim at introducing a more rigorous physics-based implicit treatment in the context of data-driven Reynolds stress models.

C. Data-driven closure modeling beyond RANS simulations

In addition to RANS modeling as reviewed above, data and machine learning have been used to provide closures for (i) the subgrid-scale (SGS) fluxes in large-eddy simulations (LESs), (ii) the interphase momentum fluxes in multiphase flow simulations [24,25], and (iii) the unresolved boundary layer physics in potential flow simulations [26]. Among these, researchers reported ill-conditioning issues in data-driven SGS models in LESs that are similar to the ill-conditioning issue in the context of RANS modeling discussed above. For example, Gamahara and Hattori [27] used a neural network to model the subgrid-scale stress in a turbulent channel flow. Compared to the predictions of Smagorinsky models, the machine learning model predicted better SGS stresses but the less satisfactory mean velocities. This observation clearly highlights the gap between *a priori* and *a posteriori* performances in assessment of turbulence models, particularly in the context of data-driven turbulence modeling. Furthermore, Durieux [28] reported that the LESs with neural network predicted SGS models become unstable if velocity-derived variables are chosen as neural network inputs, suggesting possible error amplification in the propagation of SGS stresses to mean velocities.

Note that the ill-conditioning issue only emerges if the data-driven SGS model is explicitly substituted into filtered transport equations to solve for velocities or other quantities of interest. Many other works that focus on the data-driven prediction of SGS terms (i.e., *a priori* tests) did not encounter this problem. For example, Vollant *et al.* [29,30] modeled SGS scalar flux by using neural networks based on optimal estimation. King *et al.* [31] proposed a fully adaptive self-optimizing SGS closure and demonstrated superior *a priori* performance than traditional dynamic SGS models. Maulik and San [32] trained a neural network to represent the deconvolution of flow quantities from filtered flow field. They also demonstrated excellent *a priori* performance in several canonical boundary-free flows. All these promising works have the potential of becoming data-driven SGS closures for stresses or scalar fluxes in LESs. However, a similar ill-conditioning issue in the context of RANS modeling as outlined in Sec. IB still needs to be addressed if the successes in these *a priori* tests are to be translated into *a posteriori* tests.

D. Summary

In the present work we demonstrate a systematic approach in choosing the input feature variables for machine learning in the context of turbulence modeling. Specifically, we first identify a set of vectorial or tensorial mean flow variables, e.g., strain-rate tensor, rotation-rate tensor, pressure gradient, and turbulence kinetic energy gradient. Choosing these quantities as inputs for machine learning has clear physical justifications and are supported by the practice in traditional turbulence modeling. Subsequently, we construct an invariant basis set from these variables based on the tensor representation theorem. While this approach was proposed by Ling *et al.* [5] and is not new in the present work, the application in turbulence modeling is challenging and could serve as a good illustration for researchers in many other fields where the physical quantities are described by a large number of vectors and tensors. This procedure is a clear improvement compared to earlier works with an *ad hoc* choice of many scalar variables [7,33].

Moreover, we propose a data-driven, machine-learning-based turbulence modeling framework where the Reynolds stresses are decomposed into linear and nonlinear parts and then learned

separately from DNS data. This decomposition allows implicit treatment of the linear term of the Reynolds stress, which enhances the model conditioning in solving the RANS equations for a mean velocity field without *ad hoc* blending as used in traditional turbulence modeling [22,23]. More importantly, such a decomposition clearly reflects the respective roles of linear and nonlinear terms in turbulent models. Specifically, the linear term is by far the dominant term in almost all turbulent flows in engineering practice, which partly explains the widespread use of linear eddy viscosity models in engineering computational fluid dynamics (CFD). On the other hand, the nonlinear terms can emerge as important factors in specific flows (e.g., swirling, jet impingement, and juncture flows [1,34]). Hence, treating them separately helps the machine learning algorithms distinguish these two terms. With numerical examples, we show that such a treatment enables accurate prediction of mean velocities with a data-driven Reynolds stress model.

The rest of this paper is organized as follows. Section II summarizes the machine-learning-assisted turbulence modeling framework of Wang *et al.* [7] and presents the proposed approach. Section III first highlights the ill-conditioning issues of data-driven Reynolds stress models and further demonstrates the merits of the proposed machine learning framework in *a posteriori* tests of different kinds of flows. Section IV discusses the potentials and limitations of the data-driven turbulence models. Section V summarizes the paper.

II. METHODOLOGY

Taking incompressible turbulent flows as an example, the RANS momentum equations are

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} + \nabla p - \nu \nabla^2 \mathbf{U} = \nabla \cdot \boldsymbol{\tau}, \quad (1)$$

where \mathbf{U} , p , and ν are the mean velocity, mean pressure (normalized by density), and viscosity, respectively. The Reynolds stress $\boldsymbol{\tau}$ accounts for the momentum flux due to unresolved turbulence and needs closure modeling. A turbulence model aims to close the RANS equations by constructing a mapping from the mean velocity field \mathbf{U} to the Reynolds stress field $\boldsymbol{\tau}$.

In view of the inaccuracy in RANS modeled Reynolds stresses as a critical bottleneck in the prediction accuracy of CFD simulations, Wang *et al.* [7] proposed a machine learning model for predicting Reynolds stress discrepancies by training on DNS data from similar flows. Specifically, a functional mapping $\mathbf{q} \mapsto \Delta \boldsymbol{\tau}$ from mean flow features \mathbf{q} (obtained from RANS simulations) to Reynolds stress discrepancies $\Delta \boldsymbol{\tau}$ is built by using machine learning, with the discrepancy defined as the difference between RANS predicted and DNS Reynolds stresses, i.e., $\Delta \boldsymbol{\tau} \equiv \boldsymbol{\tau}^{\text{DNS}} - \boldsymbol{\tau}^{\text{RANS}}$. Although an improved prediction of Reynolds stresses was achieved, they reported that the mean flow velocity is sensitive to the errors in Reynolds stress prediction. Since it is usually the mean velocity field and the derived quantities of interest (e.g., drag and lift) required in engineering applications, it is important to understand the error amplification in solving for the mean velocity field. The main reason is that substituting the modeled Reynolds stress explicitly into RANS equations may lead to ill-conditioned RANS equations, especially in high-Reynolds-number flows [18,21]. Moreover, the choice of mean flow features lacks a systematic procedure, which raises questions on the completeness and redundancy of the set of flow features included in the machine learning. The present work aims to address these challenges.

A. Overview of the machine learning scheme

It has been recognized that all algebraic Reynolds stress and eddy viscosity models can be written in the general form [35]

$$\begin{aligned} \mathbf{b}(\mathbf{S}, \boldsymbol{\Omega}) &= \sum_{n=1}^{10} G^{(n)} \mathcal{T}^{(n)} \\ &= G^{(1)} \mathbf{S} + G^{(2)} (\mathbf{S} \boldsymbol{\Omega} - \boldsymbol{\Omega} \mathbf{S}) + G^{(3)} (\mathbf{S}^2 - \frac{1}{3} \text{tr}(\mathbf{S}^2) \mathbf{I}) + G^{(4)} (\boldsymbol{\Omega}^2 - \frac{1}{3} \text{tr}(\boldsymbol{\Omega}^2) \mathbf{I}) + \dots, \quad (2) \end{aligned}$$

where $\text{tr}(\cdot)$ denotes the trace, \mathbf{I} denotes the identity matrix, \mathbf{b} is the deviatoric part of Reynolds stress tensor, $\{\mathcal{T}^{(n)}\}_{n=1}^{10}$ is the tensorial basis formed from strain-rate tensor $\mathbf{S} = \frac{1}{2}[\nabla\mathbf{U} + (\nabla\mathbf{U})^T]$ and rotation tensor $\Omega = \frac{1}{2}[\nabla\mathbf{U} - (\nabla\mathbf{U})^T]$, and the ellipsis denotes higher-order terms. In particular, $\mathcal{T}^{(1)} = \mathbf{S}$ and thus $G^{(1)}\mathcal{T}^{(1)}$ represents the linear part (with respect to \mathbf{S}) of the anisotropy tensor \mathbf{b} .

Inspired by this general form of algebraic Reynolds stress model, we separate the anisotropy stress tensor \mathbf{b} into linear and nonlinear parts

$$\mathbf{b} = v_t^L \mathbf{S} + \mathbf{b}^\perp, \quad (3)$$

where the linear part $v_t^L \mathbf{S}$ [coaxial with \mathbf{S} corresponds to the term $G^{(1)}\mathbf{S}$ in Eq. (2)] and \mathbf{b}^\perp represent the sum of the nonlinear terms. It is similar to the tensorial expansion of the anisotropy stress tensor \mathbf{b} in Eq. (2), while all the nonlinear terms are lumped into \mathbf{b}^\perp . The linear term can be treated implicitly to enhance the conditioning when solving the RANS equations. More details of the numerical procedure for solving the RANS equations are detailed in Appendix A. As discussed above in Sec. ID, this decomposition is more than just a numerical implicit treatment but has clear physical justifications.

In order to compute the two terms in Eq. (3) from a given Reynolds stress and strain tensor, we introduce an optimal eddy viscosity that minimizes the discrepancy between the anisotropy Reynolds stress tensor and its linear part, i.e.,

$$v_t^L = \arg \min_{v_t} \|\mathbf{b} - v_t \mathbf{S}\|, \quad (4)$$

where $\|\cdot\|$ denotes the Frobenius norm of a matrix, e.g., $\|\mathbf{S}\| = \sqrt{S_{ij}S_{ij}}$. Based on this definition, the optimal eddy viscosity v_t^L can be computed by projecting the anisotropy stress tensor on the strain rate tensor

$$v_t^L = 2 \frac{\mathbf{b} : \mathbf{S}}{\|\mathbf{S}\|\|\mathbf{S}\|}, \quad (5)$$

where $\mathbf{b} : \mathbf{S} = b_{ij}S_{ij}$ denotes a tensor double dot product.

The nonlinear term \mathbf{b}^\perp in Eq. (3) could be important even for simple shear flows in the near-wall region, since the linear term incorrectly predicts isotropic normal stresses. For more complex flows, e.g., swirling and impinging, neglecting this nonlinear term can cause the model to miss important flow physics completely. In this work we use machine learning techniques and an existing DNS database to build regression functions that predict the optimal eddy viscosity v_t^L and the nonlinear part \mathbf{b}^\perp of the anisotropy Reynolds stress tensor. In machine learning terminology, the flows used to build regression functions are referred to as the training flows and the flow to be predicted is referred to as the test flow. The detailed workflow of building these regression functions via machine learning is presented in Appendix A.

B. Construction of mean flow features as inputs of machine learning

The construction of input features is among the most critical considerations when using machine learning for physical problems. First, the choice of input and output variables must be physically motivated and justified to ensure that the function learned from the data has physical meaning. Second, the variables must be normalized properly to ensure extrapolative capabilities of the learned function. Finally, the learned function should ideally be objective with function form invariances under transformations of the coordinate system and the reference frame. Our perspective is that almost all principles that are observed in traditional turbulence modeling (see, e.g., [36]) should be equally respected in data-driven turbulence modeling. These three considerations in the present framework are presented below.

1. Physical consideration in the choice of mean flow feature variables

The general form of the nonlinear turbulent-viscosity model in Eq. (2) assumes a universal functional mapping from the strain-rate tensor \mathbf{S} and the rotation-rate tensor Ω to the Reynolds

TABLE I. Nondimensional raw mean flow variables used to construct the invariant basis. The normalized feature $\hat{\alpha}$ is obtained by normalizing the corresponding raw input α with normalization factor β according to $\hat{\alpha} = \alpha/(|\alpha| + |\beta|)$. The notation is as follows: \mathbf{U} is the mean velocity vector, k is the turbulence kinetic energy, ρ is the fluid density, ε is the turbulence dissipation rate, \mathbf{S} is the strain-rate tensor, $\mathbf{\Omega}$ is the rotation-rate tensor, $|\cdot|$ denotes vector norm, and $\|\cdot\|$ indicates the matrix norm.

Normalized raw input $\hat{\alpha}$	Description	Raw input α	Normalization factor β
$\hat{\mathbf{S}}$	strain-rate tensor	\mathbf{S}	$\frac{\varepsilon}{k}$
$\hat{\mathbf{\Omega}}$	rotation-rate tensor	$\mathbf{\Omega}$	$\ \mathbf{\Omega}\ $
$\hat{\nabla}p$	pressure gradient	∇p	$\rho D\mathbf{U}/Dt $
$\hat{\nabla}k$	TKE gradient	∇k	$\frac{\varepsilon}{\sqrt{k}}$

stress $\boldsymbol{\tau}$:

$$\boldsymbol{\tau} = \boldsymbol{\tau}(\mathbf{S}, \mathbf{\Omega}), \quad (6)$$

We note that there are at least two aspects of missing physics in this assumption. First, the turbulence is also influenced by the pressure gradient. For example, turbulence would be suppressed under a strong favorable pressure gradient [36]. Second, the general form in Eq. (6) assumes equilibrium turbulence, i.e., the turbulence production balances dissipation everywhere in the field. With such an assumption, the Reynolds stress at location \mathbf{x} only depends on the local mean velocity $\mathbf{U}(\mathbf{x})$, or more precisely its gradient $\nabla\mathbf{U}(\mathbf{x})$. However, the convection and diffusion of turbulence exist in many real applications, indicating strong nonequilibrium effects and making this single-point-based turbulent constitutive law invalid [37]. To account for the missing physics outlined above, we also include the pressure gradient ∇p and the turbulence kinetic energy (TKE) gradient ∇k in the input, leading to a more general functional mapping from mean flow quantities to the Reynolds stress

$$\boldsymbol{\tau} = g(\mathbf{S}, \mathbf{\Omega}, \nabla p, \nabla k), \quad (7)$$

where the variables of the set $\mathcal{Q} = \{\mathbf{S}, \mathbf{\Omega}, \nabla p, \nabla k\}$ are chosen as input features, which are summarized in Table I.

In addition to the tensor set \mathcal{Q} , three other features as presented in Table II are chosen from Ref. [7] to further supplement the mean flow features, all of which have clear physical interpretations. First, at the near-wall region the viscous effect becomes more important and the local Reynolds number reduces to $O(1)$. Therefore, a low-Reynolds-number turbulence model is needed for the viscous sublayer in the traditional turbulence modeling. In this work q_1 is an important indicator to inform the wall distance to the machine-learning-assisted turbulence models, leading to a data-driven low-Re

TABLE II. Supplementary mean flow features used as inputs in the regression. The normalized feature q_β is obtained by normalizing the corresponding raw feature value \hat{q}_β with normalization factor q_β^* according to $q_\beta = \hat{q}_\beta/(|\hat{q}_\beta| + |q_\beta^*|)$ except for $\beta = 1$. The notation is as follows: U_i is the mean velocity, k is the TKE, ε is the turbulence dissipation rate, \mathbf{S} is the strain rate tensor, d is the distance to the wall, and $\|\cdot\|$ indicates matrix norms. The blank entry in column 4 indicates that normalization is not necessary as the Reynolds number is nondimensional.

Feature q_β	Description	Raw feature \hat{q}_β	Normalization factor q_β^*
q_1	wall-distance based Reynolds number	$\min\left(\frac{\sqrt{k}d}{50\nu}, 2\right)$	
q_2	turbulence intensity	$\frac{k}{U_i^2}$	$\nu\ \mathbf{S}\ $
q_3	ratio of turbulent timescale to mean strain timescale	$\frac{k}{\varepsilon}$	$\frac{1}{\ \mathbf{S}\ }$

model as a counterpart of the traditional low-Re models. Second, features q_2 and q_3 carry information on the length scale and timescale of the turbulence, serving as supplements of the mean flow tensors in the set \mathcal{Q} .

2. Normalization of input features

To ensure nondimensionality of the raw inputs, the normalization scheme proposed by Ling and Templeton [33] is adopted. All raw features are normalized by local quantities, as is preferred in the practice of traditional turbulence modeling [36]. In CFD simulations these would be quantities based on the same grid point as the raw feature variables. The normalization factors for all the raw input variables are listed in Table I. Specifically, each element α in the raw input set \mathcal{Q} is normalized by a corresponding normalization factor β based on the scheme

$$\widehat{\alpha} = \frac{\alpha}{|\alpha| + |\beta|}, \quad (8)$$

which ensures that the normalized variable $\widehat{\alpha}$ falls within the range $[-1, 1]$. Note that such a normalization scheme is slightly different from that frequently used in physics and engineering, which would take the form of either $\widehat{\alpha} = \alpha/|\alpha|$ or $\widehat{\alpha} = \alpha/|\beta|$ instead. The choice in Eq. (8) is justified by the practice of machine learning where the inputs are usually normalized to the range $[-1, 1]$ or $[0, 1]$. This helps us avoid clustering of training data along certain directions within the input feature space and improves the convergence rate in the training process.

3. Invariance considerations in the choice of input features

As summarized in Table I, the raw variables for the mean flow consist of a finite tensorial set $\mathcal{Q} = \{\mathbf{S}, \boldsymbol{\Omega}, \nabla p, \nabla k\}$ with four elements, where ∇p and ∇k are transformed to antisymmetric tensors as detailed in Eq. (B1). As in traditional turbulence modeling, it is equally desirable in data-driven turbulence modeling that the trained functional form $g : (\mathbf{S}, \boldsymbol{\Omega}, \nabla p, \nabla k) \mapsto \boldsymbol{\tau}$ should be objective. That is, the function form of g should be invariant under rotational and reflectional transformations of the coordinate system or Galilean transformation (i.e., translation by a constant velocity) of the reference frame. The function form invariances associated with the rotation and reflection of the coordinate system and the Galilean transformation of the reference frame are referred to as coordinate rotational invariance, coordinate reflectional invariance, and Galilean invariance, respectively. Our formulation has rotational invariance and Galilean invariance but not reflectional invariance. However, the lack of reflectional invariance can be remedied by data augmentation, which is a standard procedure for preprocessing training data in machine learning. The three invariance properties of the present formulation and strategies to remedy the lack of reflectional invariance are examined below.

Invariance properties. If the constructed function relation $\boldsymbol{\tau} = g(\mathbf{S}, \boldsymbol{\Omega}, \nabla p, \nabla k)$ is to be valid under arbitrary rotations of the coordinate system, the relation

$$\mathbf{Q}\boldsymbol{\tau}\mathbf{Q}^T = g(\mathbf{Q}\mathbf{S}\mathbf{Q}^T, \mathbf{Q}\boldsymbol{\Omega}\mathbf{Q}^T, \mathbf{Q}\nabla p, \mathbf{Q}\nabla k) \quad (9)$$

should be satisfied [38] for any rotation matrix \mathbf{Q} , where \mathbf{Q} is an orthogonal matrix (i.e., $\mathbf{Q}^T = \mathbf{Q}^{-1}$) with determinant equal to 1. The rotational invariance of the learned function g as stated in Eq. (9) can be guaranteed by choosing invariant inputs and outputs in the learning process, specifically, by choosing the minimal integrity bases for the set $\{\mathbf{S}, \boldsymbol{\Omega}, \nabla p, \nabla k\}$ and the invariants of $\boldsymbol{\tau}$ as inputs and outputs, respectively. A minimal integrity basis is the minimal set of invariants that can represent all the polynomial invariants associated with a tensorial set under the designated transformation (rotation here). The Hilbert basis theorem states that a minimal integrity basis for a finite tensorial set has a finite number of invariants [39]. Specifically for the set $\mathcal{Q} = \{\mathbf{S}, \boldsymbol{\Omega}, \nabla p, \nabla k\}$ of second-order tensors

considered here,¹ the minimal integrity basis consists of all the traces of the independent matrix products that can be formed from the tensors according to the Cayley-Hamilton theorem [40], which amount to 47 invariants (see Table IV for details). Note that choosing the invariant tensorial bases, rather than the raw tensorial variables, as inputs and outputs of the machine learning only guarantees the rotational invariance of the learned function. Galilean invariance needs to be independently achieved by ensuring that each raw variable and its normalization factors are Galilean invariant, which is discussed below.

Galilean invariance states that the laws of motion are the same in all frames with constant velocities, which is an important prerequisite of any turbulence model [36] and should be equally satisfied by traditional or data-driven models. Therefore, the velocity *per se* is usually not a valid term in a model as mentioned in [36], because the velocity is not Galilean invariant. In contrast, the velocity gradient $\nabla \mathbf{U}$ and thus its symmetric and antisymmetric parts (\mathbf{S} and $\mathbf{\Omega}$, respectively) are all Galilean invariant and thus are valid terms to be included in a turbulence model. Similarly, the gradient of pressure ∇p and the gradient of kinetic energy ∇k are both Galilean invariant (and do not depend on the choice of reference pressure). Finally, it is straightforward to show that a term involving only Galilean invariant quantities is also Galilean invariant. Based on the general principles outlined above, the machine learning inputs and outputs in the present formulation are all Galilean invariant. Specifically, the raw inputs in Table I are all Galilean invariants since they only involve spatial gradients, e.g., \mathbf{S} and $\mathbf{\Omega}$. The raw inputs in Table II are all Galilean invariant since they only involve scalar quantities that are Galilean invariant. The normalization factors in both tables are all Galilean invariant as well. In particular, it can be shown that the normalization factor $\rho |D\mathbf{U}/Dt|$ is Galilean invariant and the details are presented in Appendix C.

Finally, the invariants associated with the antisymmetric tensors are only rotational invariants but not reflection invariants. The main motivation for using these antisymmetric tensors is that the machine learning outputs include the quantification of three-dimensional (3D) rotation of eigenvectors of stress tensors that do not have reflection invariance. The details are discussed in the following.

(i) *Motivation of the chosen outputs.* Our framework aims to augment traditional turbulence models, instead of completely replacing them, by using machine learning to predict the discrepancies between RANS modeled and true Reynolds stresses. Such discrepancies are parametrized by the scaling of the Reynolds stress tensor along its eigenvectors and the rotation of the eigenvectors. Although the scaling factors are all scalars and invariants under coordinate system transformation, the rotation of the eigenvectors does not have reflection invariance. For instance, the sign of the angle between any two vectors depends on the defined direction of the rotation axis.

(ii) *Reasons for including inputs without reflection invariance.* If we only introduce machine learning inputs with both rotational and reflection invariance, similar inputs would correspond to totally different outputs of tensor rotations (potentially with flipped signs) in the training data set. Therefore, the functional form $\mathcal{F} : \mathbf{q} \mapsto \Delta \boldsymbol{\tau}$ would be a noisy function with spikes due to the sign flipping learned from the training data. Such a machine learning model would be nonphysical with diminishing predictive capabilities.

(iii) *Consistency between outputs and inputs.* We adopted the invariants of pseudotensors constructed from the TKE gradient and the pressure gradient as a part of machine learning inputs. Similar to the invariants corresponding to a 3D rotation, the invariants of these pseudotensors are also rotational invariants but not reflection invariants. Therefore, the sign flipping of the machine learning outputs can be distinguished by different corresponding inputs and the unphysical noisy behavior of the functional form \mathcal{F} would not exist. The main purpose here is to ensure a consistent framework to predict tensor rotations, which is critical in augmenting the traditional tensor models.

¹Vectors such as ∇p and ∇k can be first transformed to the corresponding antisymmetric tensors based on Eq. (B1) in Appendix B.

Indeed, it would be more elegant to include only objective inputs and output in the machine learning, as the coordinate transformation invariance is a basic requirement in turbulence modeling. Further work is needed in identifying such formulations. It should be noted that the different conventions of coordinate system handedness would lead to different machine learning inputs even with identical training and test flows. An alternative albeit rather inefficient approach is to use training flows under both right-handed and left-handed coordinates system to augment the training data, and it has been demonstrated in [5] that the invariance can be learned by such data augmentation.

Data augmentation for achieving invariance in machine learning. In machine learning the lack of invariance property in the model to be learned from data can be remedied by data augmentation [5]. Specifically, the training data set is augmented by duplicating them in various transformed coordinate systems before performing the training. That way, the training process would be able to see the same data in almost all transformed coordinates. Consequently, only the functional forms that are valid (i.e., invariant) in all coordinate systems are learned, and any coordinate- or frame-dependent functional forms would be rejected in the training. However, depending on the invariance to be achieved through data augmentation (as detailed below), this procedure could significantly increase the amount of data and the computational costs for the training and prediction.

(i) In order to achieve reflectional invariance through data augmentation, one only needs to duplicate the data in the reflected coordinate system, which is a moderate twofold increase in the amount of data.

(ii) On the other hand, achieving three-dimensional rotational invariance requires duplicating the training data in 1000 coordinate systems, as shown by Ling *et al.* [5]. Therefore, using data augmentation to learn rotational invariance would significantly increase the computational cost and memory consumption, both of which are important considerations in machine learning.

(iii) Finally, it is not straightforward to remedy the lack of Galilean invariance with data augmentation, since the translation velocity of the reference frame is unbounded (i.e., it can be any value from $-\infty$ to ∞).

In summary, a total of 50 normalized, invariant mean flow field variables (collectively denoted by \mathbf{q}) are constructed and used as input features for the machine learning. While such a high-dimensional feature space may appear daunting even for the most experienced experts in turbulence modeling, it is not particularly challenging in the context of modern data science, as many machine learning techniques routinely handle feature spaces of thousands of dimensions or even higher [41].

C. Representation of Reynolds stress discrepancy as outputs of machine learning

Similar to choosing the inputs of machine learning, we represent the Reynolds stress discrepancies with rotationally invariant variables as the outputs of the machine learning. Following [7,42], we formulate the Reynolds stress discrepancies as six physically interpretable components (i.e., magnitude, shape, and orientation) based on the eigendecomposition of the anisotropic Reynolds stress tensor

$$\boldsymbol{\tau} = 2k\left(\frac{1}{3}\mathbf{I} + \mathbf{b}\right) = 2k\left(\frac{1}{3}\mathbf{I} + \mathbf{V}\Lambda\mathbf{V}^T\right), \quad (10)$$

where k is the turbulent kinetic energy, which indicates the magnitude of $\boldsymbol{\tau}$; \mathbf{I} is the second order identity tensor; \mathbf{b} is the deviatoric part of $\boldsymbol{\tau}$; and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$ and $\Lambda = \text{diag}[\lambda_1, \lambda_2, \lambda_3]$, with $\lambda_1 + \lambda_2 + \lambda_3 = 0$, are the orthonormal eigenvectors and eigenvalues of \mathbf{b} , respectively, indicating its shape and orientation. The eigenvalues λ_1 , λ_2 , and λ_3 are transformed to barycentric coordinates C_1 , C_2 , and C_3 and then to Cartesian coordinates ξ and η as in Refs. [7,43].

To ensure that the predicted TKE is non-negative, Wang *et al.* [7] introduce the TKE discrepancy $\Delta \log k$ as the logarithm of the ratio of the target TKE (k^*) to the RANS-simulated TKE (k^{RANS}), i.e.,

$$\Delta \log k \equiv \log \frac{k^*}{k^{\text{RANS}}}. \quad (11)$$

This is what we adopted in the present work. Finally, the unit quaternions are used to represent the transformation from the RANS eigenvectors \mathbf{V}^{RANS} to the target eigenvectors \mathbf{V}^* [44],

$$\mathbf{h} = \left[\cos \frac{\theta}{2}, n_1 \sin \frac{\theta}{2}, n_2 \sin \frac{\theta}{2}, n_3 \sin \frac{\theta}{2} \right]^T, \quad (12)$$

where $\mathbf{n} \equiv [n_1, n_2, n_3]$ defines a unique axis of unit vector and θ represents the rotation angle such that \mathbf{V}^* can be obtained via rotating \mathbf{V} by θ about the axis \mathbf{n} . This unit quaternions representation is rotational invariant and thus is preferred over the Euler angle representation used by Wang *et al.* [7]. Note that the current representation of the rotation of Reynolds stress eigenvectors is not reflectional invariant, i.e., the magnitude of each component of the unit quaternion \mathbf{h} remains the same under the reflection of frame while the sign of each component is not. As mentioned in Sec. II B, the reflection invariance of the trained machine learning model can be achieved by augmenting the training database with a reflected coordinate system.

In summary, the discrepancies ($\Delta \log k$, $\Delta \xi$, $\Delta \eta$, h_1 , h_2 , and h_3), collectively denoted by $\Delta \tau$, are used as the machine learning outputs to represent the discrepancies between the target Reynolds stress and the RANS modeled Reynolds stress. Here h_1 , h_2 , and h_3 denote the first three components of the unit quaternion \mathbf{h} . All six variables are invariant under rotations of the coordinate system. In this work, random forest regression is adopted to represent the dependence of these Reynolds stress discrepancies on a large number of scalar inputs as identified in Sec. II B. Therefore, the mean velocity and other quantities (e.g., k and ε) from RANS simulations are used to calculate the machine learning inputs, since it can be expected that the Reynolds stress discrepancies are related to the specific choice of the RANS model. Because of the dependence of the trained machine learning function on the RANS model, we recommend the usage of the same RANS model for both the training flows and the flow to be predicted.

D. Choice of machine learning algorithm and parameters

In this work, random forest regression [45] as implemented in R [46] is used to build functional mappings from the inputs (mean flow features \mathbf{q}) to the responses as identified in Sec. II A. Random forest regression is a tree-based ensemble learning method, i.e., the regression outputs are the mean prediction of individual decision trees. In this work, the regression outputs are directly used as explicit values and the detailed formulation of the random forest has no influence upon the convergence of solving mean velocity via RANS equations. An advantage of the random forest regression is that it can provide importance scores for inputs after training, which can be further used to assist the modelers to improve the existing RANS models [7]. The random forest has robust performances with only a small set of tuning parameters, which is in contrast to the commonly used neural networks [47]. The number of maximum features is set as 7, i.e., $1 + \log_2 n$, where $n = 50$ is the number of input features in this work, based on the recommendation in Ref. [45]. The number of trees is set as 300. This number is chosen by observing the out-of-bag (OOB) error to avoid possible overfitting on the training sets. We have observed the OOB error with different numbers of trees and this error is not sensitive based on our current setting of the number of trees. An example of using random forest regression to assist RANS modeling is publicly available in [48].

The computational costs of machine learning consist of the training cost and the prediction cost. The training cost depends on the amount of the training data, and it should be noted that the training procedure can be done off-line and the trained machine learning model can be applied to the prediction of other flows as well. In this work, only one training flow is used at one time and the computational cost of the training procedure is less than the corresponding standard RANS simulation. The computational cost of the prediction procedure is usually negligible compared with the cost of a typical RANS simulation. Therefore, with the trained machine-learning-assisted model, the computational cost for the baseline RANS simulation in combination with the prediction of a given flow is still much lower than LESs of the same flow.

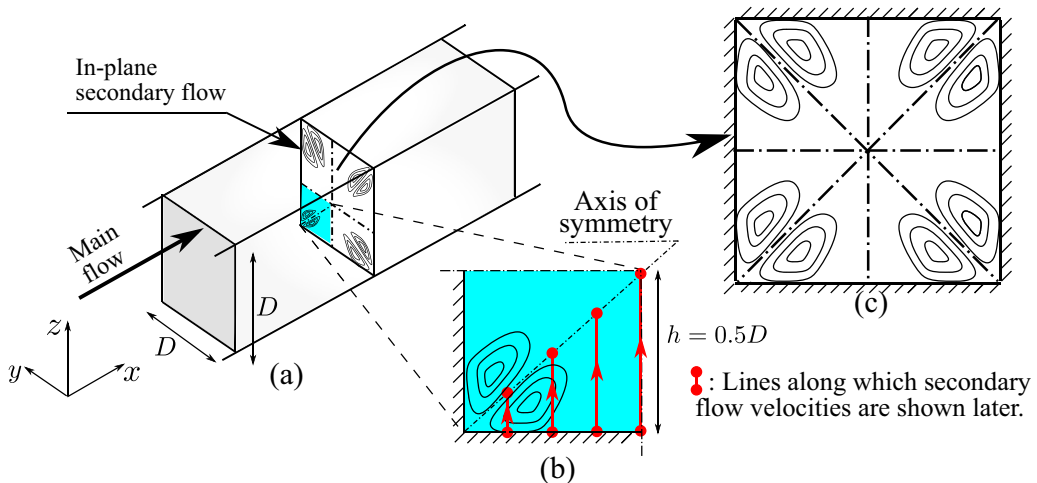


FIG. 1. (a) Computational domain for the flow in a square duct. The x coordinate represents the streamwise direction. Secondary flows induced by Reynolds stress imbalance exist in the y - z plane. (b) The computational domain covers a quarter of the cross section of the physical domain. This is due to the symmetry of the mean flow in both the y and z directions as shown in (c).

III. NUMERICAL RESULTS

Two canonical flows, the flow in a square duct and the flow over periodic hills, are investigated to evaluate the performance of the proposed method. The flow in a square duct is featured by stress-induced secondary flow and the flow over periodic hills is featured by the massive separation. These two features are challenging for traditional RANS modeling [49,50]. In this work we first perform a propagation test by using DNS Reynolds stress. The purpose of the propagation of DNS Reynolds stress to the mean velocity is to demonstrate the merit of physics-based implicit treatment. In the *a posteriori* test, the Reynolds stress field is modeled by machine learning techniques and propagated to mean velocities to evaluate the predicative capability of the proposed machine-learning-assisted turbulence modeling framework with physics-based implicit treatment.

A. Case setup

A schematic of the flow in a square duct is presented in Fig. 1 to show the physical domain and the computational domain. A two-dimensional simulation is performed, since the flow is fully developed along the streamwise direction. In addition, the computational domain covers only a quarter of the cross section, as shown in Fig. 1(b), due to the symmetry of the flow along the y and z directions. All lengths are normalized by the height of the computational domain $h = 0.5D$, where D is the height of the duct. The Reynolds number Re is based on the height of the computational domain h and bulk velocity U_b . The Launder-Gibson Reynolds stress transport model [51] is used for the RANS simulations of both the training flow and the test flow. Direct numerical simulation data at $Re = 2200$ and 3500 are obtained from Pinelli *et al.* [52]. Experimental data at $Re = 125\,000$ is obtained from Gessner and Emery [53].

Another training-prediction case with the flows over periodic hills is shown in Fig. 2. The test flow is the flow over periodic hills at $Re = 5600$ [50]. The geometry of the computational domain of the test flow is shown in Fig. 2. The training flow has a steeper hill profile indicated by the dashed line in Fig. 2. The Reynolds number Re is based on the crest height H and the bulk velocity U_b at the crest. Periodic boundary conditions are applied in the streamwise (x) direction and nonslip boundary conditions are applied at the walls. The baseline RANS simulations used the Launder-Sharma k - ε model [54].

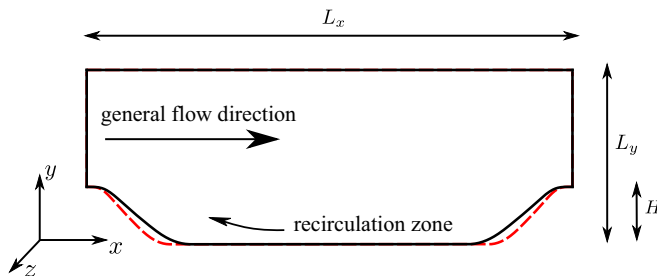


FIG. 2. Computational domain for the flow over periodic hills. The solid line indicates the configuration of the test flow and the dashed line indicates the configuration of the training flow. The hill width of the training flow is 0.8 of the hill width of the test flow. The x , y , and z coordinates are aligned in the streamwise, wall-normal, and spanwise directions, respectively.

All the RANS simulations are performed in an open-source CFD platform OpenFOAM, using a built-in steady-state incompressible flow solver `simpleFoam` [55], in which the SIMPLE algorithm [56] is used. In the RANS simulations, the y^+ of the first cell center is kept less than 1 and thus no wall model is applied.

The training-prediction cases in the *a posteriori* test are summarized in Table III. Case 1 is investigated to show the Reynolds-number extrapolation for which DNS data are available to examine the prediction performance in detail. Case 2 is chosen to demonstrate the capability of the proposed framework at higher Reynolds number where only experimental data are available. Case 3 is studied to demonstrate the prediction performance for which the training flow and the test flow have different geometrical configurations.

B. Propagation of DNS Reynolds stresses

In this test the DNS Reynolds stresses are used to illustrate the merit of physics-based implicit treatment. Three types of turbulence models are compared: Reynolds stress models with explicit treatment, linear eddy viscosity models, and RSMs with implicit treatment. The explicit treatment means that the modeled Reynolds stress is directly substituted into the RANS equations to solve for mean velocity as an explicit term. The dependence of Reynolds stress upon the strain rate can still be taken into account by updating the modeling of Reynolds stress during the time stepping (or iterations for steady problems). However, merely updating the Reynolds stress explicitly based on the solved mean velocity would not improve the conditioning of the RANS equations, which is further discussed in [21]. The implicit treatment means that the modeled Reynolds stress implicitly depends on the strain rate through an optimized eddy viscosity. Such an implicit treatment would improve the conditioning of RANS equations since the optimized eddy viscosity has impact upon the coefficient matrix of the discretized RANS equations and thus influences the condition number.

In the test of Reynolds stress models, the unclosed term in the momentum equation is replaced with the DNS Reynolds stress. In the test of linear eddy viscosity models, the eddy viscosity term is replaced with the optimal eddy viscosity ν_t^L obtained from DNS data. Specifically, the optimal eddy viscosity ν_t^L is computed by minimizing the discrepancy between the linear eddy viscosity model

TABLE III. Training-prediction scenarios in the *a posteriori* test.

Cases	Training set	Test set
1	flow in a square duct at $Re = 2200$ [52]	flow in a square duct at $Re = 3500$ [52]
2	flow in a square duct at $Re = 2200$ [52]	flow in a square duct at $Re = 125\,000$ [53]
3	flow over periodic hills at $Re = 5600$ (steeper hill profile)	flow over periodic hills at $Re = 5600$ [50]

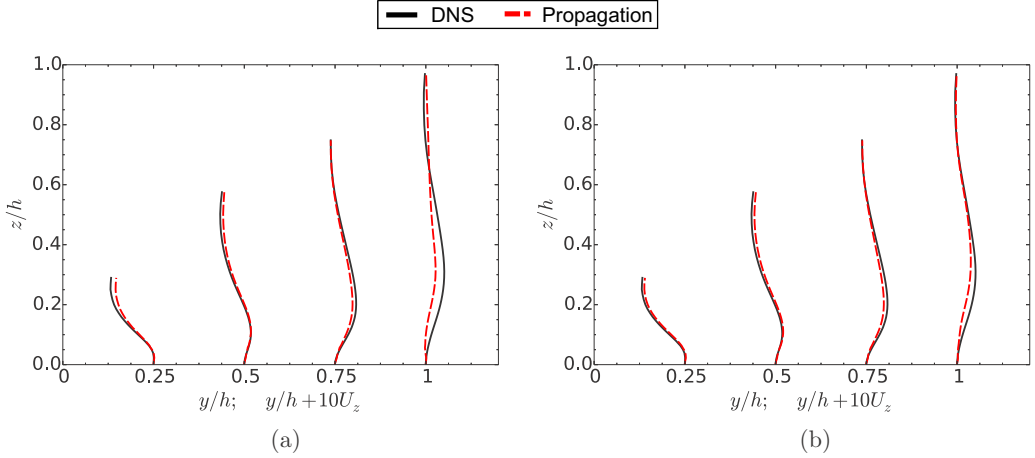


FIG. 3. Comparison of secondary flow velocity U_z by using (a) the Reynolds stress model with explicit treatment and (b) the Reynolds stress model with implicit treatment. The results obtained using the linear eddy viscosity model capture no secondary flow and are thus omitted here.

and the DNS Reynolds stress data, i.e., $v_t^L = \arg \min_{v_t} \|\mathbf{b}^{\text{DNS}} - v_t \mathbf{S}^{\text{DNS}}\|$. Compared with the eddy viscosity models, the RSM with implicit treatment further takes into account the nonlinear part of Reynolds stress, i.e., $\boldsymbol{\tau} = v_t^L \mathbf{S} + (\boldsymbol{\tau}^{\text{DNS}} - v_t^L \mathbf{S}^{\text{DNS}})$. Two canonical flows, the flow in a square duct and the flow over periodic hills, are studied to compare the performance of these three types of models. This propagation test demonstrates that the linear eddy viscosity model is unreliable within the region where the misalignment between the Reynolds stress tensor and strain-rate tensor is not negligible. In the test with explicit treatment of Reynolds stress, the DNS data are employed to demonstrate the problem of the machine learning modeling approach under an ideal scenario. Specifically, this ideal scenario means that the initial mean velocity field is already the same as the DNS mean velocity and the modeled Reynolds stress is also the same as the DNS Reynolds stress. Such an ideal scenario represents an absolute performance ceiling of any machine-learning-assisted modeling approach with the explicit treatment. Under this ideal scenario, we demonstrate that a small error in DNS Reynolds stress (e.g., sampling error) can lead to a relatively large error in the solved mean velocity field. If the modeling of Reynolds stress is further updated by evaluating the data-driven model with the solved mean velocity, the errors in the modeled Reynolds stress and the solved mean velocity would be further amplified as has been demonstrated for the plane channel flow in [21]. Therefore, an implicit treatment is needed to improve the conditioning of the machine-learning-assisted RANS modeling.

1. Flow in a square duct

The secondary velocity U_z based on DNS Reynolds stress is shown in Fig. 3 by using Reynolds stress models with explicit and implicit treatments. It can be seen in Fig. 3(a) that the solved secondary velocity U_z agrees well with DNS data within most regions, except along the symmetry plane $y/h = 1$. Such good agreement of solved secondary velocity indicates that the Reynolds stress models lead to well-conditioned discretized momentum equations for the flow in a square duct. Similar quality of secondary velocity can be obtained by using the RSM with implicit treatment. On the other hand, the linear eddy viscosity models cannot predict the secondary flow due to the limitation in representing the normal stress imbalance, the results of which are omitted here for simplicity.

2. Flow over periodic hills

Although the Reynolds stress models with explicit treatment perform well for the flow in a square duct, they are potentially unreliable since it can lead to pronounced errors in mean velocity even for

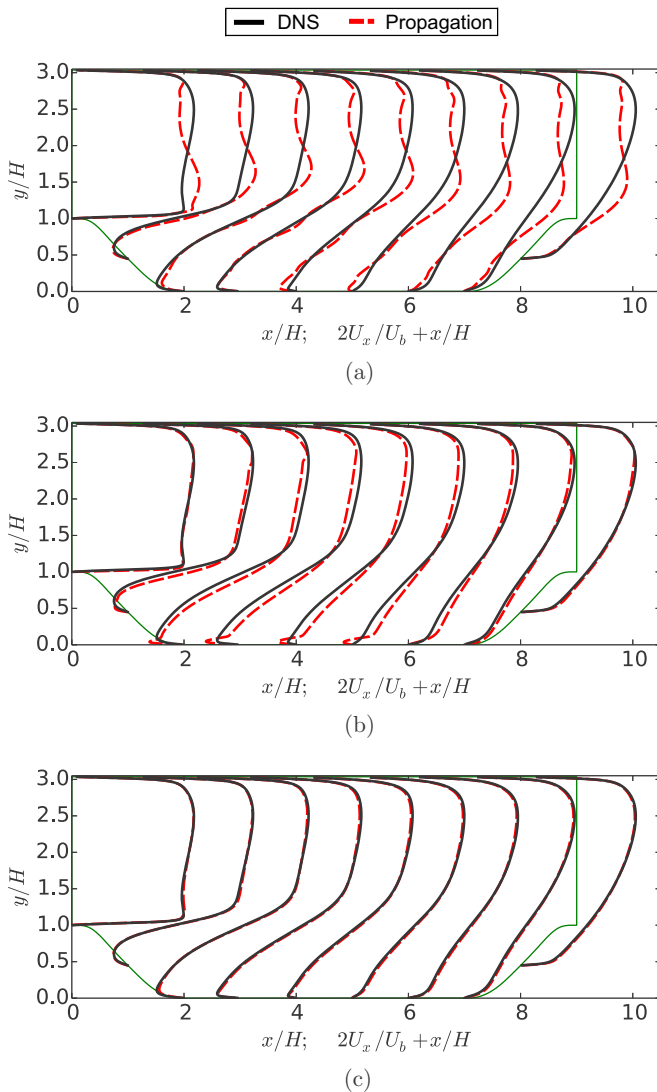


FIG. 4. Solved mean velocity field for the flow over periodic hills at $Re = 5600$ by using (a) the Reynolds stress model with an explicit treatment, (b) the eddy viscosity model, and (c) the Reynolds stress model with implicit treatment. The DNS data are utilized as the modeled term to represent the best possible performance among the respective classes of models.

the turbulent channel flows [18,21]. In this work we demonstrate the issue of Reynolds stress models with explicit treatment for the flow with massive separation. Figure 4(a) shows that the solved mean velocity field does not agree with the DNS data by using the Reynolds stress model with explicit treatment for the flow over periodic hills. It should be noted that the velocity shown in Fig. 4(a) is obtained by using DNS Reynolds stress and not the modeled Reynolds stress. Therefore, the unsatisfactory results in Fig. 4(a) indicate the best possible performance of machine-learning-assisted turbulence modeling via directly substituting the modeled Reynolds stress into RANS equations.

Better results of mean velocity can be achieved, as shown in Fig. 4(b), by using linear eddy viscosity models than Reynolds stress models with explicit treatment. However, noticeable differences can still be observed between the solved mean velocity and the DNS data. The main reason is that the

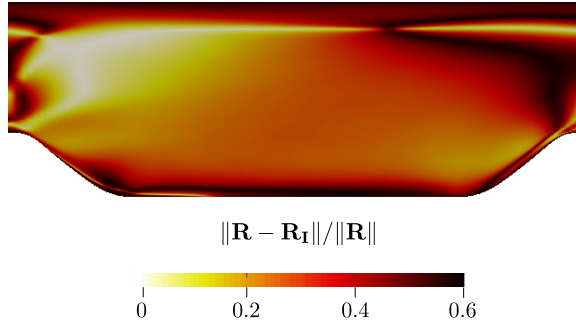


FIG. 5. Indicator of the deficiency of the linear eddy viscosity model due to the misalignment of eigenvectors between the Reynolds stress tensor and strain rate tensor.

misalignment between the Reynolds stress tensor and the strain rate tensor is neglected by using linear eddy viscosity models. Such misalignment can be quantified by the rotation matrix \mathbf{R} from the eigenvectors of the Reynolds stress tensor to the ones of the strain rate tensor. Figure 5 presents an indicator of misalignment calculated based on the deviation of the rotation matrix \mathbf{R} from the identity matrix \mathbf{R}_I .

Unlike the eddy viscosity models, the RSMs with implicit treatment take into account the difference between the linear part of Reynolds stress τ^L and the true Reynolds stress τ . The purpose is to address the misalignment of eigenvectors between the Reynolds stress tensor and the strain rate tensor. It can be seen in Fig. 4(c) that the solved mean velocity field has much better agreement with DNS data than with the results obtained by using the Reynolds stress models with explicit treatment in Fig. 4(a) and the eddy viscosity models in Fig. 4(b). By using DNS Reynolds stress data as the ideal machine-learning-modeled stress, the propagation test results in Fig. 4 demonstrate the superiority of RSMs with implicit treatment in achieving the predicative capability of the mean velocity field.

C. *A posteriori* test

1. Flow in a square duct

In the first case, the random forest is trained by using the flow in a square duct at Reynolds number $Re = 2200$. The flow at Reynolds number $Re = 3500$ is used as the test flow. It should be noted that the DNS data at $Re = 3500$ are only used to evaluate the machine learning prediction and not for training the machine learning model. The baseline RANS result indicates the results obtained from standard RANS simulations. The machine learning results are denoted by ML in the legends of figures for simplicity. It can be seen in Fig. 6 that the baseline RANS simulated normal components of Reynolds stress qualitatively capture the imbalance between τ_{yy} and τ_{zz} . However, the simulated normal stress imbalance is noticeably greater than the DNS data, especially around the near-wall region. Such greater normal stress imbalance between τ_{yy} and τ_{zz} explains the stronger secondary flow of baseline RANS simulation. Compared with the baseline RANS simulated stress components, the machine-learning-predicted normal stress components τ_{yy} and τ_{zz} demonstrate much better agreement with the DNS data in Fig. 6.

The linear part of Reynolds stress τ^L is predicted and presented in Fig. 7. It can be seen that the normal components τ_{yy}^L and τ_{zz}^L of the linear part of the Reynolds stress are similar to each other. The main reason is that the linear part of the Reynolds stress is obtained by projecting the DNS Reynolds stress onto the strain rate tensor and neglecting the nonlinear parts of the DNS Reynolds stress. Therefore, the linear part of the Reynolds stress follows the eddy viscosity assumption and thus would have no normal stress imbalance. It can be seen in Fig. 7 that the machine-learning-predicted linear part of the Reynolds stress shows good agreement with the linear part of the Reynolds stress obtained from DNS data.

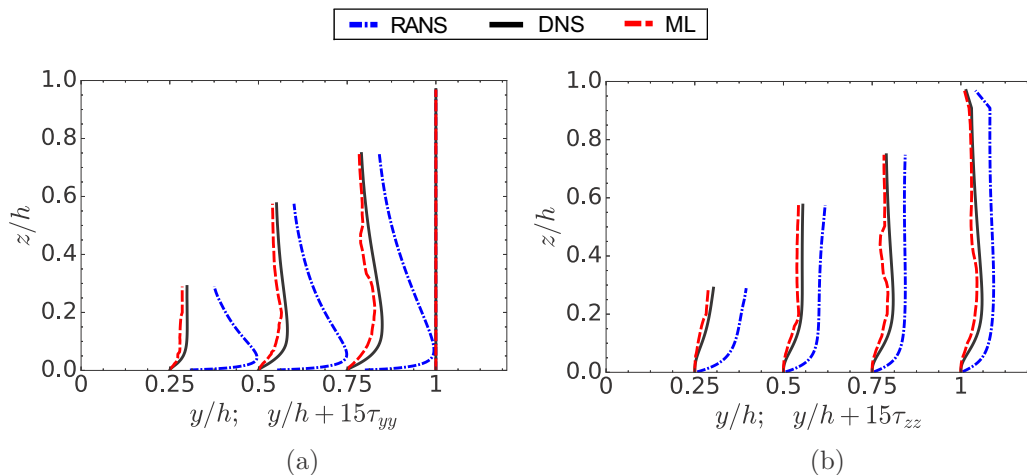


FIG. 6. Prediction of DNS Reynolds stress components of the flow in a square duct at Reynolds number $Re = 3500$, including (a) τ_{yy}^{DNS} and (b) τ_{zz}^{DNS} at $y/h = 0.25, 0.5, 0.75$, and 1 . The training flow is at Reynolds number $Re = 2200$.

With both the satisfactory machine learning prediction of Reynolds stress in Fig. 6 and its linear part in Fig. 7, it can be expected that the nonlinear term $\tau^\perp = \tau - \tau^L$ would have good agreement with the nonlinear part of the DNS Reynolds stress. The comparison of the nonlinear Reynolds stress term is presented in Fig. 8. It can be seen that the noticeable negative stress τ_{yy}^\perp can be seen near both the sidewall and the bottom wall of the duct based on the DNS data. However, the negative nonlinear stress is overpredicted near the side wall and underpredicted near the bottom wall for the RANS simulation. Compared with the RANS results, the machine learning prediction in Fig. 8 demonstrates much better agreement with the pattern of DNS data.

In addition to the nonlinear part of the Reynolds stress, the optimal eddy viscosity ν_t^l is also needed in solving for the mean velocity. It can be seen in Fig. 9 that the eddy viscosity is close to zero in

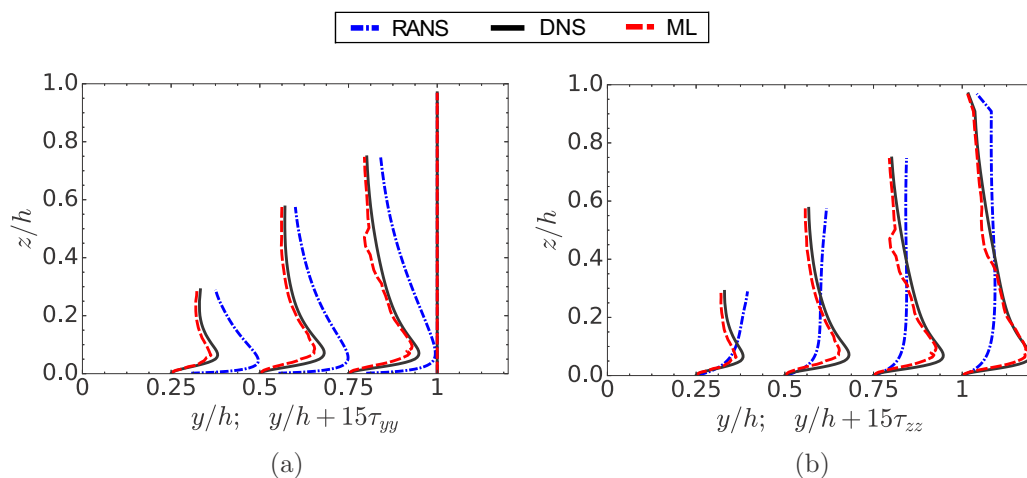


FIG. 7. Prediction of the linear part of Reynolds stress components of the flow in a square duct at Reynolds number $Re = 3500$, including (a) τ_{yy}^L and (b) τ_{zz}^L at $y/h = 0.25, 0.5, 0.75$, and 1 . The training flow is at Reynolds number $Re = 2200$.

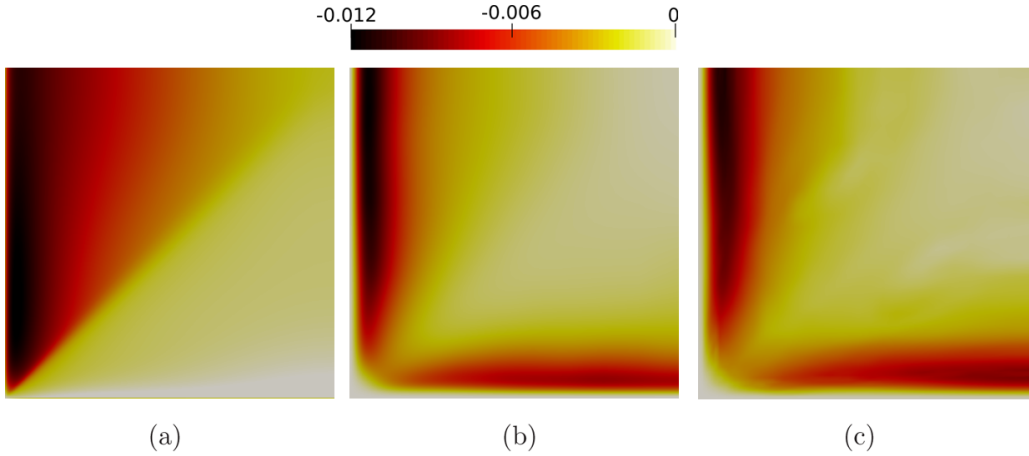


FIG. 8. Nonlinear part of the Reynolds stress τ_{yy}^{\perp} in a square duct at Reynolds number $Re = 3500$, including (a) RANS simulated results, (b) DNS data, and (c) prediction of RSMs with implicit treatment. The training flow is at Reynolds number $Re = 2200$. The color of the contour denotes the value of the stress component; the light color here indicates small magnitude.

the near-wall region and increases towards the diagonal of the duct. The machine-learning-predicted eddy viscosity agrees well with the DNS eddy viscosity at most regions in Fig. 9. It should be noted that a few noticeable differences can be observed between the machine-learning-predicted eddy viscosity and the DNS data, e.g., along $y/h = 0.75$ and $y/h = 1$. However, the velocity gradient is relatively small within these regions and thus such a difference has little influence in solving for the mean velocity field as demonstrated in Fig. 11.

By substituting the machine-learning-predicted eddy viscosity and nonlinear part of the Reynolds stress into RANS equations, the mean velocity field is solved and presented in Fig. 10. It can be seen in Fig. 10(a) that the RANS simulated secondary flow penetrates too much toward the left bottom region (corner region between the perpendicular walls). The machine-learning-predicted secondary flow in Fig. 10(c) demonstrates better agreement with the DNS data in this corner region. In addition, the

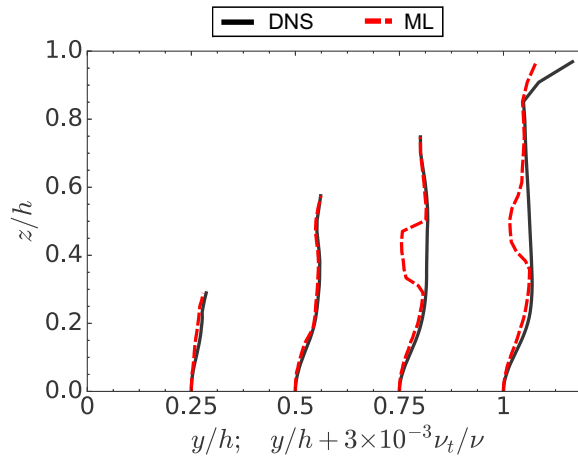


FIG. 9. Optimal eddy viscosity v_t^{\perp} of the flow in a square duct at Reynolds number $Re = 3500$ at $y/h = 0.25$, 0.5 , 0.75 , and 1 . The training flow is at Reynolds number $Re = 2200$.

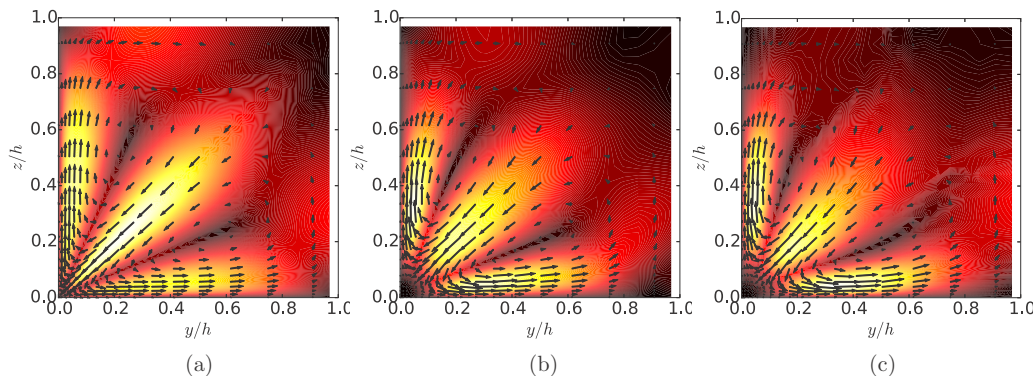


FIG. 10. Secondary flow fields in a square duct at Reynolds number $Re = 3500$, including (a) RANS simulated results, (b) DNS data, and (c) prediction of the RSM with implicit treatment. The training flow is at Reynolds number $Re = 2200$. The color of the contour denotes the magnitude of the secondary flow. The light color here indicates large magnitude.

shape and location of the secondary vortex are better predicted by our machine learning framework as shown in Fig. 10(c).

Four profiles of secondary flow are presented in Fig. 11 for a more quantitative comparison of the secondary flow prediction. It can be seen that the baseline RANS simulated mean velocity field overestimates the magnitude of the secondary flow, especially around the corner region. On the other hand, the mean velocity based on the RSM with implicit treatment shows much better agreement with the DNS data. However, it should be noted that the test flow is at Reynolds number $Re = 3500$, close to that of the training flow ($Re = 2200$). Therefore, the satisfactory predictive capability as demonstrated in Fig. 11 does not necessarily guarantee a similar performance at a higher Reynolds number.

To further demonstrate the general applicability of the proposed framework, we employ the same training flow and investigate another test flow, i.e., the flow in a square duct at a much higher Reynolds number $Re = 1.25 \times 10^5$. We evaluate the prediction performance of the proposed framework by using the experimental data of this test flow along the vertical axis of symmetry and along the diagonal of the square duct [53]. The inverse flow near the bottom of the axis of symmetry is not captured

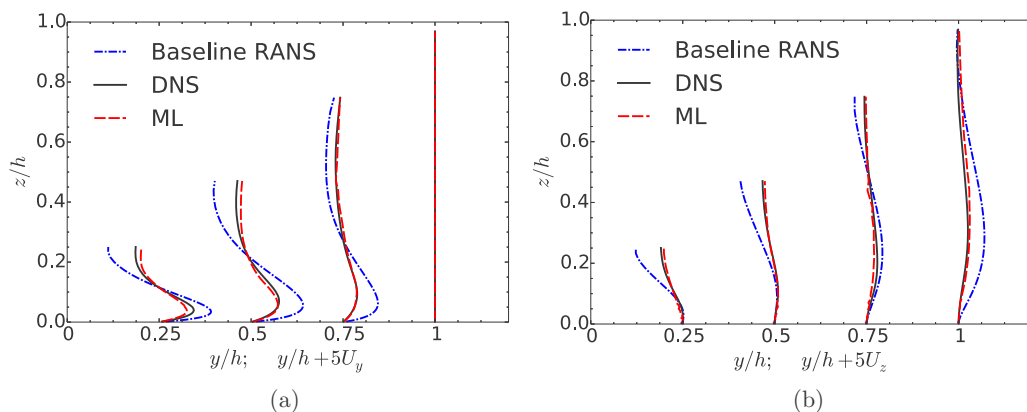


FIG. 11. Secondary flow in a square duct at Reynolds number $Re = 3500$ predicted by the RSM with implicit treatment at $y/h = 0.25, 0.5, 0.75$, and 1 including (a) U_y and (b) U_z . The training flow is at Reynolds number $Re = 2200$.

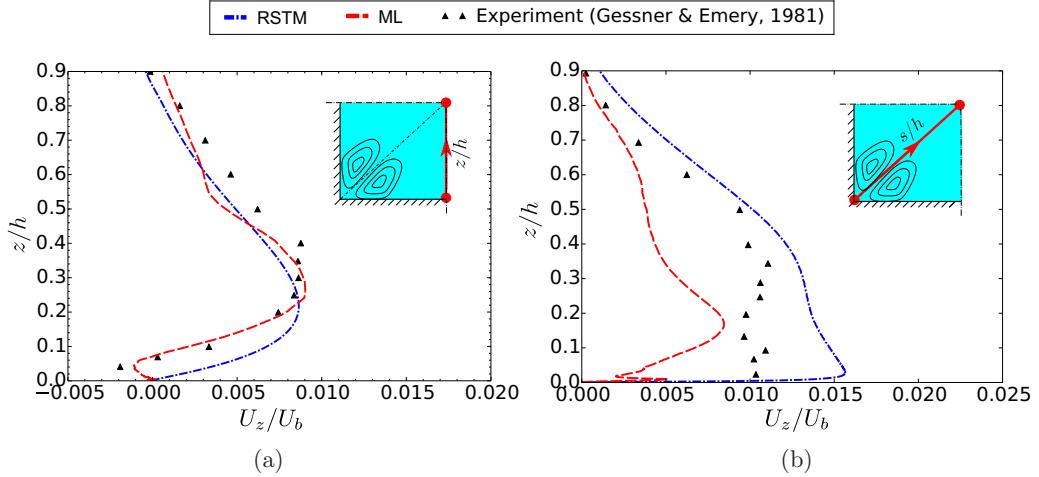


FIG. 12. Comparison of predicted secondary velocity U_z at Reynolds number $Re = 1.25 \times 10^5$ with experimental data [53] (denoted by \blacktriangle). Comparisons are shown along (a) the vertical axis of symmetry and (b) the diagonal of the duct. The training flow is at Reynolds number $Re = 2200$.

by the baseline RANS simulation in Fig. 12(a). This inverse flow in experimental data indicates that there is a small vortex with an opposite rotation direction around this region, in addition to the main vortex of the secondary flow. The failure of predicting the inverse flow in Fig. 12(a) means that this small vortex is completely missing in the baseline RANS simulation results. In contrast, this missing flow characteristic is successfully captured by the machine learning prediction. It should be noted that this flow characteristic is not observed in the training flow at a much lower Reynolds number $Re = 2200$. One reason for the successful prediction of the small inverse flow is the formulation of our data-driven augmentation framework, i.e., it is the Reynolds stress discrepancies but not the whole Reynolds stress that is predicted by the machine learning framework. Specifically, Wu *et al.* [12] calibrated the Reynolds stress discrepancies at a lower Reynolds number $Re \approx 5000$ and applied the calibrated discrepancies to correct the RANS simulation at a higher Reynolds number $Re = 125000$. They reported that the trend of inverse flows could be reproduced even though the Reynolds stress discrepancies are calibrated at a much lower Reynolds number. Therefore, the trend of small inverse flows can be predicted even if the machine learning model is overfitted for the training database at a lower Reynolds number. However, it should be noted that the inverse flow was noticeably underestimated in the work by Wu *et al.* [12], and the machine learning prediction in this work achieves better agreement with the experimental data. Therefore, the successful prediction of the inverse flow in Fig. 12(a) is strong evidence that the machine-learning-assisted turbulence modeling indeed has the potential of revealing the physics within the data, rather than simply interpolating with the available data.

As shown in Fig. 12(b), the secondary flow U_z is overpredicted by baseline RANS simulation along the diagonal of the duct. In contrast, the mean velocity U_z obtained by the proposed framework underestimates the magnitude of the secondary flow in Fig. 12(b). Although there is no significant improvement of results along the diagonal, it should be noted that the machine learning prediction in Fig. 12(b) indeed corrects the baseline RANS simulated results towards the right direction.

2. Flow over periodic hills

In the cases investigated above, the training flow and the test flow share the same geometrical configuration and only differ in Reynolds numbers. In order to demonstrate the capability of the proposed framework for the flows with different geometries, we further study the flows over periodic hills with a different shape of hill profiles. The training flow is the flow over periodic hill at $Re = 5600$

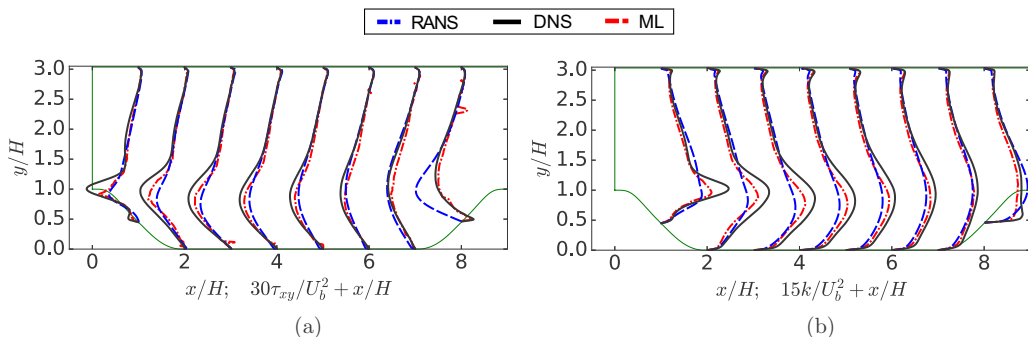


FIG. 13. Prediction of (a) the shear stress component τ_{xy} of the Reynolds stress tensor and (b) the TKE at $x/H = 1, 2, \dots, 8$. The test flow is the flow over a periodic hill at $Re = 5600$. The training flow is at the same Reynolds number but has a steeper hill profile, as shown in Fig. 2.

[50] and the test flow is also at $Re = 5600$ but has a steeper hill profile as described in Sec. III A. It can be seen in Fig. 13(a) that the RANS simulation underpredicts the shear stress τ_{xy} downstream of the hill crest. Such underprediction of shear stress is mainly due to the underprediction of the TKE of the RANS simulation as shown in Fig. 13(b). Compared with the results of RANS simulation, the machine learning prediction shows better agreement with DNS data for both the shear stress and the TKE. Although such improvement is limited at $x/H = 1$, it still better predicts the magnitude of Reynolds stress than RANS simulation as shown in Fig. 13. The improvement of the machine learning prediction further downstream becomes more prominent.

The linear part of the Reynolds stress τ^L is also predicted. It can be seen in Fig. 14 that the linear part of the Reynolds stress is similar to the DNS Reynolds stress in Fig. 13 except for several regions, e.g., the region around the bottom wall and the region within the upper channel around $y/H = 2.5$. Therefore, the nonlinear part of the DNS Reynolds stress is expected to be less dominant in most regions for the flow over periodic hills. The prediction of the proposed framework shows good agreement with the linear part of the Reynolds stress obtained from DNS data. Similar to the prediction of the Reynolds stress, the machine learning prediction of the linear part of the Reynolds stress shows less noticeable improvement at $x/H = 1$.

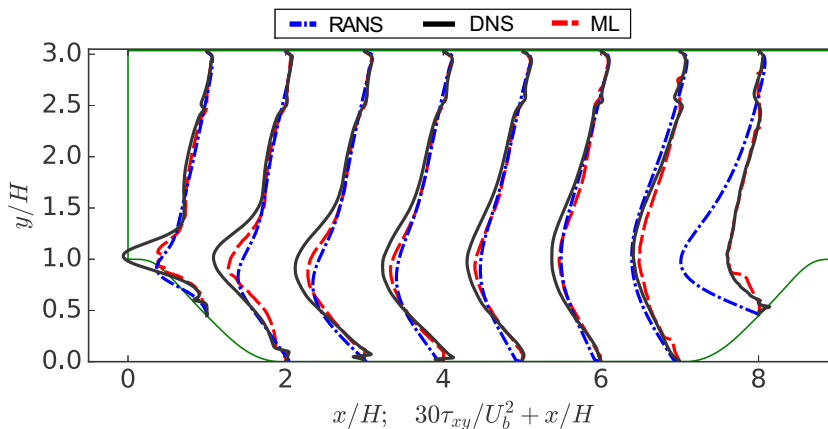


FIG. 14. Prediction of the shear stress component τ_{xy} of the linear part of Reynolds stress at $x/H = 1, 2, \dots, 8$. The test flow is the flow over a periodic hill at $Re = 5600$. The training flow is at the same Reynolds number but has a steeper hill profile, as shown in Fig. 2.

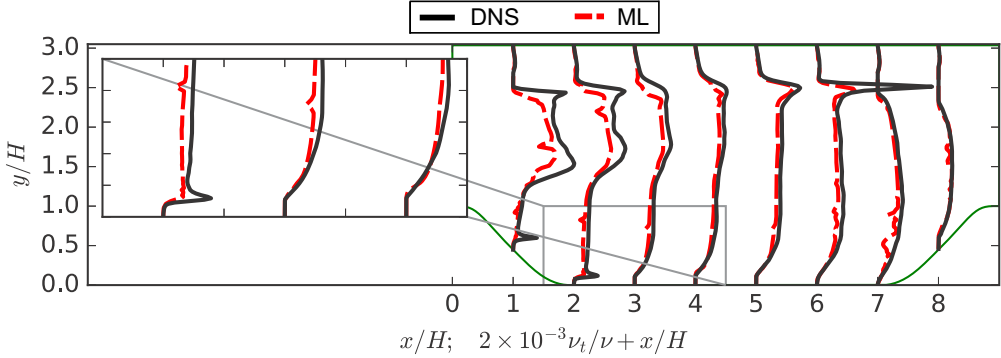


FIG. 15. Machine-learning-predicted optimal eddy viscosity at $x/H = 1, 2, \dots, 8$. The test flow is the flow over a periodic hill at $Re = 5600$. The training flow is at the same Reynolds number but has a steeper hill profile, as shown in Fig. 2.

In addition to the improvement shown in the prediction of the DNS Reynolds stress and its linear part, the prediction of the optimal eddy viscosity also demonstrates improvement as shown in Fig. 15. It can be seen in Fig. 15 that the machine-learning-predicted eddy viscosity has good agreement with the DNS eddy viscosity, except for a few regions where the DNS eddy viscosity changes rapidly. Such deterioration in the performance of machine learning prediction is expected, since the functions with such behavior pose more difficulties in machine learning. However, the peak value of eddy viscosity at these regions is usually not important for solving for mean velocity. This is because the strain rate tensor is close to zero in these regions, corresponding to the peak value of eddy viscosity.

The comparison of the mean velocity field in Fig. 16 shows that the mean velocity obtained by the proposed framework has better agreement with the DNS data. Specifically, the reverse flow extends to $x/H = 4$ in the DNS data, defining the size of the separation bubble downstream of the hill crest. The RANS simulation results indicate that the reverse flow ends approximately around $x/H = 3$, which significantly underestimates the size of the separation bubble. The magnitude of the velocity at the upper channel region is also underpredicted by the RANS simulation results from $x/H = 1$ to $x/H = 5$. Compared with the RANS simulation results, the machine learning prediction provides more accurate reverse flow, especially from $x/H = 1$ to $x/H = 3$. Although overprediction of reverse flow can be observed in the prediction of the proposed framework from $x/H = 3$ to $x/H = 4$, the separation region is still better predicted than RANS simulation results.

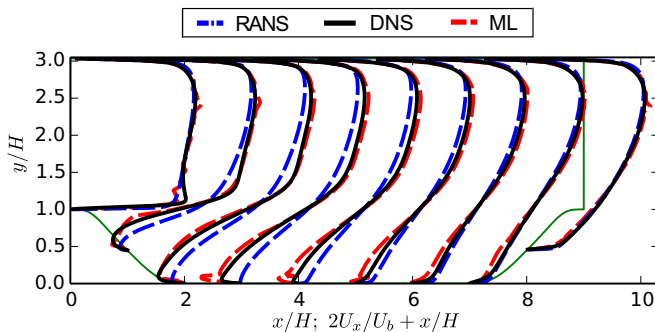


FIG. 16. Streamwise velocity by the RSM with implicit treatment at $x/H = 1, 2, \dots, 8$. The test flow is the flow over a periodic hill at $Re = 5600$. The training flow is at the same Reynolds number but has a steeper hill profile, as shown in Fig. 2.

In addition, the proposed framework achieves a much better prediction of the mean velocity than the RANS simulation results at the upper channel region.

IV. DISCUSSION: POTENTIALS AND LIMITATIONS OF DATA-DRIVEN TURBULENCE MODELS

The present work addresses the ill-conditioning issue in a class of data-driven turbulence models that aim to model the Reynolds stresses with machine learning. Therefore, it is helpful to provide a broad yet brief view of data-driven turbulence modeling, which has emerged as a promising yet controversial subject in the past few years. A more comprehensive state-of-the-art overview is presented elsewhere [57]. Summarized briefly, three distinctly different approaches to data-driven turbulence modeling have been pursued by different groups.

(i) Weatheritt and Sandberg [15,16] used gene expression programming to develop algebraic Reynolds stress models based on symbolic regression.

(ii) Duraisamy and co-workers [14,58] used machine learning to predict discrepancies in the source terms in the turbulence transport equations for existing models (e.g., the Spalart-Allmaras model, the $k-\omega$ model, or Reynolds stress models).

(iii) Wang *et al.* [7] and Ling *et al.* [6] used machine learning to directly predict the Reynolds stresses or their discrepancies compared to the truth.

A widely accepted yardstick used in the turbulence modeling community to assess turbulence models is that an ideal model should demonstrate robust predictive capabilities in a wide range of flows without flow-specific tuning and be interpretable, ideally in explicit analytical forms, which are referred to as universality and interpretability requirements, respectively, hereafter. The three approaches above are discussed in light of each of these requirements.

A. Universality of turbulence models

In terms of the universality requirement, the existing traditional turbulence models are still far from satisfactory. Most existing data-driven turbulence models, including the one presented in our work, are still in their infancy and have shown only limited predictive capabilities, typically in flows that are close to the training flows. That is, the training and prediction flows belong to the same class of flows (e.g., massively separated flows) but with variations of flow configurations such as geometry or Reynolds number. However, their predictive capabilities may improve as more experiences are accumulated and the methodologies are refined later on. The machine learning inputs and outputs in this work are all rotational invariants and Galilean invariants. Therefore, the trained model is objective under any rotation transformation and Galilean transformation. Although several inputs and outputs are not reflection invariants, the associated extrapolation capability can be achieved by reflecting the coordinates system to augment the training data. However, the extrapolation capability still depends on the diversity of the training data. By using the training database of flows with a specific characteristic, e.g., recirculation or stress-induced secondary flow, we have demonstrated that an unknown flow with similar characteristics can be predicted. We envision that the extrapolation capability can be extended to more complex flows with a more diversified and systematically generated high-fidelity simulation database. In addition, the extrapolation capability for more complex flows can also be achieved by automatically classifying the flow field into different regions and making a prediction for each region accordingly. On the other hand, if we take a less ambitious perspective and consider the data-driven models (particularly the second and third approaches) as augmentation of traditional models through a data-driven, flow-specific correction, then their universality become less critical. For example, in practical applications, a data-driven model could turn itself off when the part of the flow is not present in the training data set and is too “far” from the flows in the data. That is, the data-driven model can revert to the traditional baseline turbulence model that is used. To this end, Wu *et al.* [59] demonstrated a fully automatic, statistically rigorous way of measuring the “distance” between two flows (or any parts thereof) based on the mean

flow fields obtained from RANS simulations. With such a distance metric, it is possible to ensure that a machine-learning-augmented model is at least better than or the same as the corresponding baseline model it utilizes.

B. Interpretability of turbulence models

Regarding the interpretability requirement, it may appear that the symbolic regression approach specifically aims to find Reynolds stress models that are in analytical forms, while the second and third approaches based on machine learning (neural networks and random forests) do not produce models in analytical forms and are not amenable to interpretation. However, the delineation between these approaches may not be as clear as it first appears. First, although the analytical Reynolds stress model learned from one class of flows [15] may be rather simple and indeed has similar complexity to existing advanced turbulence models, this may not necessarily be true for other flows. For example, a model learned from a diverse data set from different flows may be an analytical expression with the number of terms too large for human comprehension, which may still be difficult to interpret despite its analytical nature. Second, if one is willing to compromise the predictive performance, a neural network based model (used in the second and third approaches) may be regularized (e.g., by using Ridge and LASSO [60]) to yield simple architecture and thus becomes more interpretable. Finally, both neural networks and the random forest can provide the importance of input mean flow features (see, e.g., the discussion in [7]), which may help traditional model developers incorporate additional variables into existing models [61]. Therefore, the fundamental differences among the three approaches above lie in the classical trade-off between predictive capability and interpretability in models [60] and not in the specific forms of the chosen model (e.g., symbolic regression vs neural networks or random forests).

Ultimately, both universality and interpretability requirements are intimately related to the fundamental question in turbulence modeling: Does there exist a universal turbulent constitutive relation? Generations of researchers have labored for many decades on dozens of turbulence models, yet none of them achieved predictive generality, which seems to indicate that the answer is “no.” If so, then flow-specific tuning and fudge factors would be inevitable if good predictive performances are desirable. The machine-learning-based turbulence models can be considered automatic, flow-specific tuning schemes based on the flow regime to be predicted and the flow regimes that present in the training database.

V. CONCLUSION

While earlier works demonstrated the capabilities of machine learning in predicting improved Reynolds stresses, obtaining an improved mean velocity field remains a challenge of machine-learning-assisted turbulence modeling. The main reason is the sensitivity of the mean velocity with regard to the errors in the prediction of Reynolds stress. In this work we propose a physics-based implicit treatment to model Reynolds stress by using machine learning techniques. Specifically, the optimal eddy viscosity and the nonlinear part of the Reynolds stress are both predicted. In the propagation test, the DNS Reynolds stress is used in solving for the mean velocity to illustrate the ideal scenario of machine-learning-assisted turbulence modeling approaches. The propagation test with DNS Reynolds stress shows that satisfactory mean velocity can be achieved by the Reynolds stress models with implicit treatment. In the *a posteriori* test, three training-prediction cases are investigated to demonstrate the predictive capability of the proposed framework. In the first and second cases, a machine learning model is trained on the flow in a square duct at Reynolds number $Re = 2200$ and the flows in a square duct at Reynolds numbers $Re = 3500$ and $Re = 1.25 \times 10^5$ are predicted. In the third case, a machine learning model is trained on the flow over periodic hills at Reynolds number $Re = 5600$ and the flow with a steeper hill profile is predicted. The satisfactory prediction performance of the mean velocity field demonstrates the predictive capability of the proposed machine-learning-assisted framework. Specifically, the machine-learning-trained

model successfully predicts the mean flow pattern in the second case that is not even shown in the training flow. It provides strong evidence that machine-learning-assisted turbulence modeling can reveal flow physics from the existing data, instead of merely fitting on the existing data. With the capability in predicting the mean velocity field, the proposed physics-based implicit treatment leads to a practical machine-learning-assisted turbulence modeling framework in real applications where the mean velocity field and other quantities of interests need to be predicted.

ACKNOWLEDGMENTS

The authors would like to thank the two reviewers for their constructive and valuable comments, which helped improve the quality and clarity of this paper. In particular, the authors would like to thank the comments on Galilean invariance from one of the reviewers.

Parts of this paper have been presented in [62,63]. We gratefully acknowledge the discussion and collaborations from Dr. Jianxun Wang and Dr. Julia Ling in the early stage of this research.

APPENDIX A: DETAILED ALGORITHMS OF THE DATA-DRIVEN REYNOLDS STRESS MODEL

The detailed work flow of constructing a data-driven Reynolds stress model with implicit treatment and use it for solving the RANS equations are presented as follows. First, we write Eq. (3) in terms of the Reynolds stress tensor rather than the anisotropy tensor

$$\mathbf{b} = \nu_t^L \mathbf{S} + (\mathbf{b} - \nu_t^L \mathbf{S}) = \nu_t^L \mathbf{S} + \{\boldsymbol{\tau} - [\nu_t^L \mathbf{S} + \text{tr}(\boldsymbol{\tau})]\} = \nu_t^L \mathbf{S} + (\boldsymbol{\tau} - \boldsymbol{\tau}^L), \quad (\text{A1})$$

where $\text{tr}(\boldsymbol{\tau})$ represents the trace of the Reynolds stress tensor and $\boldsymbol{\tau}^L$ denotes the linear part of the Reynolds stress tensor.

The procedure of the proposed framework is as follows.

(i) Perform baseline RANS simulations on both the training flows and the test flow to obtain mean flow features \mathbf{q} and the Reynolds stress tensor $\boldsymbol{\tau}^{\text{RANS}}$.

(ii) Train regression functions for Reynolds stress discrepancies $\Delta\boldsymbol{\tau}$ and $\Delta\boldsymbol{\tau}^L$ and use the trained regression functions to predict the test flow.

(a) Compute the discrepancies fields $\Delta\boldsymbol{\tau} = \boldsymbol{\tau}^{\text{DNS}} - \boldsymbol{\tau}^{\text{RANS}}$ and $\Delta\boldsymbol{\tau}^L = \boldsymbol{\tau}^L - \boldsymbol{\tau}^{\text{RANS}}$ for the training flows based on the DNS data and construct regression functions $f_1 : \mathbf{q} \mapsto \Delta\boldsymbol{\tau}$ and $f_2 : \mathbf{q} \mapsto \Delta\boldsymbol{\tau}^L$ by using the physics-informed machine learning framework [7].

(b) Use trained regression functions f_1 and f_2 to predict the discrepancies fields $\Delta\boldsymbol{\tau}$ and $\Delta\boldsymbol{\tau}^L$ for the test flow and compute the corresponding Reynolds stress fields $\boldsymbol{\tau}$ and $\boldsymbol{\tau}^L$ by adding the predicted discrepancies fields to the RANS simulated Reynolds stress $\boldsymbol{\tau}^{\text{RANS}}$.

(iii) Train regression functions for eddy viscosity ν_t^L and use the trained regression functions to predict for the test flow.

(a) Compute the least-squares eddy viscosity ν_t^L for the training flows based on the DNS data and construct regression function $f_3 : \mathbf{q} \mapsto \nu_t^L$ by using machine learning.

(b) Use trained regression functions f_3 to predict the least-squares eddy viscosity ν_t^L for the test flow.

(iv) Substitute the predicted ν_t^L , $\boldsymbol{\tau}$, and $\boldsymbol{\tau}^L$ into RANS equations for the anisotropy stress tensor as shown in Eq. (A1) and solve for the corresponding mean velocity field.

In this workflow, the linear part and nonlinear part of the Reynolds stress are predicted separately. For the linear part, we only predict the optimal eddy viscosity ν_t^L and thus treat the linear part implicitly. To obtain the nonlinear part, we predict both the Reynolds stress itself and the linear part of Reynolds stress and we calculate the nonlinear part of the Reynolds stress by subtracting the linear part of the Reynolds stress from the Reynolds stress itself.

Solving for the mean velocity is performed in a finite-volume CFD platform OpenFOAM, using a modified flow solver that allows the implicit treatments of Reynolds stress predicted by the machine learning model. Specifically, the modified flow solver is based on a built-in steady-state incompressible flow solver `simpleFoam` [55], in which the SIMPLE algorithm [56] is used. Unlike

TABLE IV. Minimal integrity basis for the symmetric tensor $\widehat{\mathbf{S}}$ and antisymmetric tensors $\widehat{\mathbf{\Omega}}$, $\widehat{\mathbf{A}}_p$, and $\widehat{\mathbf{A}}_k$. In the implementation, $\widehat{\mathbf{S}}$ is the strain-rate tensor, $\widehat{\mathbf{\Omega}}$ is the rotation-rate tensor, $\widehat{\mathbf{A}}_p$ and $\widehat{\mathbf{A}}_k$ are the antisymmetric tensors associated with pressure gradient $\widehat{\nabla}p$ and the gradient of turbulence kinetic energy $\widehat{\nabla}k$, and n_S and n_A denote the numbers of symmetric and antisymmetric raw tensors for the integrity basis. The invariant bases listed in column 3 are the trace of each tensor listed. The asterisk on a term indicates that all terms formed by cyclic permutation of antisymmetric tensor labels (e.g., $\widehat{\mathbf{\Omega}}^2\widehat{\mathbf{A}}_p\widehat{\mathbf{S}}^*$ is short for $\widehat{\mathbf{\Omega}}^2\widehat{\mathbf{A}}_p\widehat{\mathbf{S}}$ and $\widehat{\mathbf{A}}_p^2\widehat{\mathbf{\Omega}}\widehat{\mathbf{S}}$) are also included.

(n_S, n_A)	Feature index	Invariant bases
(1, 0)	1–2	$\widehat{\mathbf{S}}^2, \widehat{\mathbf{S}}^3$
(0, 1)	3–5	$\widehat{\mathbf{\Omega}}^2, \widehat{\mathbf{A}}_p^2, \widehat{\mathbf{A}}_k^2$
(1, 1)	6–14	$\widehat{\mathbf{\Omega}}^2\widehat{\mathbf{S}}, \widehat{\mathbf{\Omega}}^2\widehat{\mathbf{S}}^2, \widehat{\mathbf{\Omega}}^2\widehat{\mathbf{S}}\widehat{\mathbf{\Omega}}\widehat{\mathbf{S}}^2,$ $\widehat{\mathbf{A}}_p^2\widehat{\mathbf{S}}, \widehat{\mathbf{A}}_p^2\widehat{\mathbf{S}}^2, \widehat{\mathbf{A}}_p^2\widehat{\mathbf{S}}\widehat{\mathbf{A}}_p\widehat{\mathbf{S}}^2,$ $\widehat{\mathbf{A}}_k^2\widehat{\mathbf{S}}, \widehat{\mathbf{A}}_k^2\widehat{\mathbf{S}}^2, \widehat{\mathbf{A}}_k^2\widehat{\mathbf{S}}\widehat{\mathbf{A}}_k\widehat{\mathbf{S}}^2$
(0, 2)	15–17	$\widehat{\mathbf{\Omega}}\widehat{\mathbf{A}}_p, \widehat{\mathbf{A}}_p\widehat{\mathbf{A}}_k, \widehat{\mathbf{\Omega}}\widehat{\mathbf{A}}_k$
(1, 2)	18–41	$\widehat{\mathbf{\Omega}}\widehat{\mathbf{A}}_p\widehat{\mathbf{S}}, \widehat{\mathbf{\Omega}}\widehat{\mathbf{A}}_p\widehat{\mathbf{S}}^2, \widehat{\mathbf{\Omega}}^2\widehat{\mathbf{A}}_p\widehat{\mathbf{S}}^*, \widehat{\mathbf{\Omega}}^2\widehat{\mathbf{A}}_p\widehat{\mathbf{S}}^2*, \widehat{\mathbf{\Omega}}^2\widehat{\mathbf{S}}\widehat{\mathbf{A}}_p\widehat{\mathbf{S}}^2*,$ $\widehat{\mathbf{\Omega}}\widehat{\mathbf{A}}_k\widehat{\mathbf{S}}, \widehat{\mathbf{\Omega}}\widehat{\mathbf{A}}_k\widehat{\mathbf{S}}^2, \widehat{\mathbf{\Omega}}^2\widehat{\mathbf{A}}_k\widehat{\mathbf{S}}^*, \widehat{\mathbf{\Omega}}^2\widehat{\mathbf{A}}_k\widehat{\mathbf{S}}^2*, \widehat{\mathbf{\Omega}}^2\widehat{\mathbf{S}}\widehat{\mathbf{A}}_k\widehat{\mathbf{S}}^2*,$ $\widehat{\mathbf{A}}_p\widehat{\mathbf{A}}_k\widehat{\mathbf{S}}, \widehat{\mathbf{A}}_p\widehat{\mathbf{A}}_k\widehat{\mathbf{S}}^2, \widehat{\mathbf{A}}_p^2\widehat{\mathbf{A}}_k\widehat{\mathbf{S}}^*, \widehat{\mathbf{A}}_p^2\widehat{\mathbf{A}}_k\widehat{\mathbf{S}}^2*, \widehat{\mathbf{A}}_p^2\widehat{\mathbf{S}}\widehat{\mathbf{A}}_k\widehat{\mathbf{S}}^2*$
(0, 3)	42	$\widehat{\mathbf{\Omega}}\widehat{\mathbf{A}}_p\widehat{\mathbf{A}}_k$
(1, 3)	43–47	$\widehat{\mathbf{\Omega}}\widehat{\mathbf{A}}_p\widehat{\mathbf{A}}_k\widehat{\mathbf{S}}, \widehat{\mathbf{\Omega}}\widehat{\mathbf{A}}_k\widehat{\mathbf{A}}_p\widehat{\mathbf{S}}, \widehat{\mathbf{\Omega}}\widehat{\mathbf{A}}_p\widehat{\mathbf{A}}_k\widehat{\mathbf{S}}^2, \widehat{\mathbf{\Omega}}\widehat{\mathbf{A}}_k\widehat{\mathbf{A}}_p\widehat{\mathbf{S}}^2, \widehat{\mathbf{\Omega}}\widehat{\mathbf{A}}_p\widehat{\mathbf{S}}\widehat{\mathbf{A}}_3\widehat{\mathbf{S}}^2$

the standard `simpleFoam` solver, the modified flow solver uses the machine-learning-predicted v_i^L , $\boldsymbol{\tau}$, and $\boldsymbol{\tau}^L$ to represent the modeled Reynolds stress $\boldsymbol{\tau}^m$, i.e., $\boldsymbol{\tau}^m = v_i^L\mathbf{S} + (\boldsymbol{\tau} - \boldsymbol{\tau}^L) + \text{tr}(\boldsymbol{\tau})$. The strain rate tensor \mathbf{S} is treated implicitly in the modified flow solver. For numerical discretizations of the RANS equations, the second-order central difference scheme is chosen for all terms except for the convection term, which is discretized with a second-order upwind scheme.

For the boundary conditions of the machine-learning-predicted v_i^L , $\boldsymbol{\tau}$, and $\boldsymbol{\tau}^L$ in this work, the fixed zero values are applied at the walls and periodic boundary conditions are applied in the streamwise (x) direction. The boundary conditions of the mean velocity and the pressure are treated the same as the standard RANS simulations, i.e., periodic boundary conditions are applied in the streamwise (x) direction and nonslip boundary conditions are applied at the walls.

APPENDIX B: INTEGRITY BASIS OF MEAN FLOW FEATURES

The minimal integrity bases for rotational invariance with a given input set

$$\widehat{\mathcal{Q}} = \{\widehat{\mathbf{S}}, \widehat{\mathbf{\Omega}}, \widehat{\nabla}p, \widehat{\nabla}k\}$$

of tensors are shown in Table IV. Spencer [39,40] provided a systematic procedure of generating minimal invariant bases from a set of symmetric and antisymmetric tensors. In order to utilize this procedure, we need to first transform the vectors $\widehat{\nabla}p$ and $\widehat{\nabla}k$ to the corresponding antisymmetric tensors by using the mapping [40]

$$\widehat{\mathbf{A}}_p = -\mathbf{I} \times \widehat{\nabla}p, \quad (\text{B1a})$$

$$\widehat{\mathbf{A}}_k = -\mathbf{I} \times \widehat{\nabla}k, \quad (\text{B1b})$$

where \mathbf{I} is the second-order identity tensor and the symbol \times denotes tensor cross product.

APPENDIX C: GALILEAN INVARIANCE

In Sec. II B 3, we stated that all the feature variables in Tables I and II and their corresponding normalization factors are Galilean invariant. This is evident from the fact that most of the variables and normalization factors contain only terms associated with the velocity gradient $\nabla \mathbf{U}$ (e.g., \mathbf{S} and Ω), pressure gradient ∇p , and TKE gradient ∇k . However, the Galilean invariance of the normalization factor $\rho|Du/Dt|$ is not evident. In this appendix we show that the material derivative of velocity is Galilean invariant.

The mean velocity at location \mathbf{x} and time t observed in a stationary reference frame is denoted by $\mathbf{U}(\mathbf{x}, t)$. The mean velocity observed in a reference frame moving with constant velocity \mathbf{C} can be written as $\mathbf{U}^*(\mathbf{x}^*, t) = \mathbf{U}(\mathbf{x} - \mathbf{C}t, t) + \mathbf{C}$, where $\mathbf{x}^* = \mathbf{x} - \mathbf{C}t$ represents the spatial location observed in the moving reference frame. The material derivative of the velocity U_j^* in the moving reference frame is derived as follows:

$$\frac{\partial U_i^*}{\partial t} = \frac{\partial U_i}{\partial t} - C_j \frac{\partial U_i^*}{\partial x_j^*}, \quad U_j^* \frac{\partial U_i^*}{\partial x_j^*} = (U_j + C_j) \frac{\partial U_i^*}{\partial x_j^*}. \quad (\text{C1})$$

Combining the two terms in Eq. (C1) and utilizing the fact that $\partial U_i^*/\partial x_j^* = \partial U_i/\partial x_j$ (because the reference frame velocity is constant) yields

$$\frac{\partial U_i^*}{\partial t} + U_j^* \frac{\partial U_i^*}{\partial x_j^*} = \frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j}, \quad (\text{C2})$$

that is,

$$\frac{DU_i^*}{Dt} = \frac{DU_i}{Dt}, \quad (\text{C3})$$

which demonstrates that the material derivative of velocity \mathbf{U} is invariant under Galilean transformation. The merit of ensuring Galilean invariance is that the trained machine learning model $g(\mathbf{U})$ is valid in all inertial frames, i.e., $g(\mathbf{U}) = g(\mathbf{U} + \mathbf{C})$, enhancing the generality of the trained model.

It is noted that several input features in the works of Wang *et al.* [7] and Ling and Templeton [33] are not Galilean invariant. For example, the raw features of pressure gradient along streamline $U_i \partial P/\partial x_i$ and the ratio of convection to production of turbulent kinetic energy $U_i dk/dx_i$, and the normalization factor $U_i U_i$ for turbulence intensity and the normalization factor $\rho \partial U_i^2/\partial x_i$ for the ratio of pressure normal stresses to shear stresses. Therefore, the machine learning framework in this work is expected to achieve better extrapolation capability with the Galilean invariance for all input and output quantities.

-
- [1] T. Craft, B. Launder, and K. Suga, Development and application of a cubic eddy-viscosity model of turbulence, *Int. J. Heat Fluid Flow* **17**, 108 (1996).
 - [2] T. A. Oliver and R. D. Moser, Bayesian uncertainty quantification applied to RANS turbulence models, *J. Phys.: Conf. Ser.* **318**, 042032 (2011).
 - [3] B. Tracey, K. Duraisamy, and J. Alonso, *Proceedings of the 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Dallas, 2013* (AIAA, Reston, 2013), paper 2013-0259.
 - [4] B. Tracey, K. Duraisamy, and J. J. Alonso, *Proceedings of the 53rd AIAA Aerospace Sciences Meeting, Kissimmee, 2015* (AIAA, Reston, 2015), paper 2015-1287.
 - [5] J. Ling, R. Jones, and J. Templeton, Machine learning strategies for systems with invariance properties, *J. Comput. Phys.* **318**, 22 (2016).
 - [6] J. Ling, A. Kurzawski, and J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, *J. Fluid Mech.* **807**, 155 (2016).
 - [7] 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).

- [8] E. Dow and Q. Wang, *Proceedings of the 52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Denver, 2011* (AIAA, Reston, 2011), paper 2011-1762.
- [9] E. J. Parish and K. Duraisamy, A paradigm for data-driven predictive modeling using field inversion and machine learning, *J. Comput. Phys.* **305**, 758 (2016).
- [10] A. P. Singh and K. Duraisamy, Using field inversion to quantify functional errors in turbulence closures, *Phys. Fluids* **28**, 045110 (2016).
- [11] H. Xiao, J.-L. Wu, J.-X. Wang, R. Sun, and C. J. Roy, Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier-Stokes equations: A data-driven, physics-based, Bayesian approach, *J. Comput. Phys.* **324**, 115 (2015).
- [12] 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).
- [13] K. Duraisamy, Z. J. Zhang, and A. P. Singh, *Proceedings of the 53rd AIAA Aerospace Sciences Meeting, Kissimmee, 2015* (Ref. [4]), paper 2015-1284.
- [14] A. P. Singh, S. Medida, and K. Duraisamy, Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils, *AIAA J.* **55**, 2215 (2017).
- [15] J. Weatheritt and R. Sandberg, A novel evolutionary algorithm applied to algebraic modifications of the RANS stress-strain relationship, *J. Comput. Phys.* **325**, 22 (2016).
- [16] J. Weatheritt and R. Sandberg, The development of algebraic stress models using a novel evolutionary algorithm, *Int. J. Heat Fluid Flow* **68**, 298 (2017).
- [17] 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).
- [18] R. L. Thompson, L. E. B. Sampaio, F. A. 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).
- [19] S. Poroseva (private communication).
- [20] M. Lee and R. D. Moser, Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$, *J. Fluid Mech.* **774**, 395 (2015).
- [21] J. Wu, H. Xiao, R. Sun, and Q. Wang, RANS equations with Reynolds stress closure can be ill-conditioned, [arXiv:1803.05581](https://arxiv.org/abs/1803.05581).
- [22] B. Basara and S. Jakirlic, A new hybrid turbulence modelling strategy for industrial CFD, *Int. J. Numer. Methods Fluids* **42**, 89 (2003).
- [23] R. Maduta and S. Jakirlic, Improved RANS computations of flow over the 25°-slant-angle Ahmed body, *SAE Int. J. Passeng. Cars-Mech. Syst.* **10**, 649 (2017).
- [24] 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).
- [25] 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).
- [26] A. N. Marques, Q. Wang, and Y. Marzouk, Data-driven integral boundary-layer modeling for airfoil performance prediction in laminar regime, *AIAA J.* **56**, 482 (2018).
- [27] M. Gamahara and Y. Hattori, Searching for turbulence models by artificial neural network, *Phys. Rev. Fluids* **2**, 054604 (2017).
- [28] T. Durieux, Exploring the use of artificial neural network based subgrid scale models in a variational multiscale formulation, Master's thesis, Delft University of Technology, 2015.
- [29] A. Volland, G. Balarac, G. Geraci, and C. Corre, *Studying Turbulence Using Numerical Simulation Databases - XV*, Proceedings of the 2014 Summer Program, Center for Turbulence Research (Stanford University, Stanford, 2014).
- [30] A. Volland, G. Balarac, and C. Corre, Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures, *J. Turbul.* **18**, 854 (2017).
- [31] R. N. King, P. E. Hamlington, and W. J. Dahm, Autonomic closure for turbulence simulations, *Phys. Rev. E* **93**, 031301 (2016).
- [32] R. Maulik and O. San, A neural network approach for the blind deconvolution of turbulent flows, *J. Fluid Mech.* **831**, 151 (2017).

- [33] 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).
- [34] D. Apsley and M. Leschziner, Investigation of advanced turbulence models for the flow in a generic wing-body junction, *Flow Turbul. Combust.* **67**, 25 (2001).
- [35] S. Pope, A more general effective-viscosity hypothesis, *J. Fluid Mech.* **72**, 331 (1975).
- [36] P. R. Spalart, Philosophies and fallacies in turbulence modeling, *Prog. Aeronaut. Sci.* **74**, 1 (2015).
- [37] J. Lumley, Toward a turbulent constitutive relation, *J. Fluid Mech.* **41**, 413 (1970).
- [38] C. G. Speziale, S. Sarkar, and T. B. Gatski, Modelling the pressure-strain correlation of turbulence: An invariant dynamical systems approach, *J. Fluid Mech.* **227**, 245 (1991).
- [39] A. Spencer and R. Rivlin, Isotropic integrity bases for vectors and second-order tensors, *Arch. Ration. Mech. Anal.* **9**, 45 (1962).
- [40] R. W. Johnson, *Handbook of Fluid Dynamics* (CRC, Boca Raton, 2016).
- [41] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, Berlin, 2006).
- [42] M. Emory, J. Larsson, and G. Iaccarino, Modeling of structural uncertainties in Reynolds-averaged Navier-Stokes closures, *Phys. Fluids* **25**, 110822 (2013).
- [43] S. Banerjee, R. Krahl, F. Durst, and C. Zenger, Presentation of anisotropy properties of turbulence, invariants versus eigenvalue approaches, *J. Turbul.* **8**, N32 (2007).
- [44] J. Wu, R. Sun, S. Laizet, and H. Xiao, Representation of Reynolds stress perturbations with application in machine-learning-assisted turbulence modeling, [arXiv:1709.05683](https://arxiv.org/abs/1709.05683).
- [45] L. Breiman, Random forests, *Mach. Learn.* **45**, 5 (2001).
- [46] R. Ihaka and R. Gentleman, R: A language for data analysis and graphics, *J. Comput. Graph. Stat.* **5**, 299 (1996).
- [47] A. Liaw and M. Wiener, Classification and regression by randomForest, *R news* **2**, 18 (2002).
- [48] <https://github.com/xiaoh/turbulence-modeling-PIML>.
- [49] A. Huser and S. Biringen, Direct numerical simulation of turbulent flow in a square duct, *J. Fluid Mech.* **257**, 65 (1993).
- [50] M. Breuer, N. Peller, C. Rapp, and M. Manhart, Flow over periodic hills: Numerical and experimental study in a wide range of Reynolds numbers, *Comput. Fluids* **38**, 433 (2009).
- [51] M. Gibson and B. Launder, Ground effects on pressure fluctuations in the atmospheric boundary layer, *J. Fluid Mech.* **86**, 491 (1978).
- [52] A. Pinelli, M. Uhlmann, A. Sekimoto, and G. Kawahara, Reynolds number dependence of mean flow structure in square duct turbulence, *J. Fluid Mech.* **644**, 107 (2010).
- [53] F. Gessner and A. Emery, The numerical prediction of developing turbulent flow in rectangular ducts, *J. Fluids Eng.* **103**, 445 (1981).
- [54] B. Launder and B. Sharma, Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc, *Lett. Heat Mass Transfer* **1**, 131 (1974).
- [55] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, *Comput. Phys.* **12**, 620 (1998).
- [56] S. Patankar, *Numerical Heat Transfer and Fluid Flow* (CRC, Boca Raton, 1980).
- [57] K. Duraisamy, G. Iaccarino, and H. Xiao, Turbulence modeling in the age of data, [arXiv:1804.00183](https://arxiv.org/abs/1804.00183).
- [58] A. P. Singh, R. Matai, K. Duraisamy, and P. Durbin, *Proceedings of the 23rd AIAA Computational Fluid Dynamics Conference, Denver, 2017* (AIAA, Reston, 2017), paper 2017-3626.
- [59] J.-L. Wu, J.-X. Wang, H. Xiao, and J. Ling, A priori assessment of prediction confidence for data-driven turbulence modeling, *Flow, Turbul. Combust.* **99**, 25 (2017).
- [60] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning* (Springer, Berlin, 2013), Vol. 112.
- [61] P. Durbin (private communication).
- [62] J.-X. Wang, J.-L. Wu, J. Ling, G. Iaccarino, and H. Xiao, *Studying Turbulence Using Numerical Simulation Databases - XVI, Proceedings of the 2016 Summer Program, Center for Turbulence Research* (Stanford University, Stanford, 2016).
- [63] J.-X. Wang, J. Wu, J. Ling, G. Iaccarino, and H. Xiao, A comprehensive physics-informed machine learning framework for predictive turbulence modeling, [arXiv:1701.07102](https://arxiv.org/abs/1701.07102).