

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/350995103>

An interpretable framework of data-driven turbulence modeling using deep neural networks

Article in Physics of Fluids · April 2021

DOI: 10.1063/5.0048909

CITATION
1

READS
890

6 authors, including:



[Chao Jiang](#)
Harbin Institute of Technology

9 PUBLICATIONS 21 CITATIONS

[SEE PROFILE](#)



[Ricardo Vinuesa](#)
KTH Royal Institute of Technology

190 PUBLICATIONS 1,972 CITATIONS

[SEE PROFILE](#)



[Junyi Mi](#)
Harbin Institute of Technology

4 PUBLICATIONS 10 CITATIONS

[SEE PROFILE](#)



[Shujin Laima](#)
38 PUBLICATIONS 366 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Large-scale atmospheric flow interaction with ice sheets [View project](#)



Vision-based data anomaly detection for SHM [View project](#)

An interpretable framework of data-driven turbulence modeling using deep neural networks

Cite as: Phys. Fluids **33**, 055133 (2021); <https://doi.org/10.1063/5.0048909>

Submitted: 27 February 2021 . Accepted: 19 April 2021 . Published Online: 27 May 2021

 Chao Jiang (姜超),  Ricardo Vinuesa, Ruilin Chen (陈瑞林), Junyi Mi (米俊亦),  Shujin Laima (赖马树金), and  Hui Li (李惠)

COLLECTIONS

 This paper was selected as Featured



[View Online](#)



[Export Citation](#)



[CrossMark](#)

Physics of Fluids
SPECIAL TOPIC: Tribute to
Frank M. White on his 88th Anniversary

SUBMIT TODAY!

An interpretable framework of data-driven turbulence modeling using deep neural networks

Cite as: Phys. Fluids **33**, 055133 (2021); doi: [10.1063/5.0048909](https://doi.org/10.1063/5.0048909)

Submitted: 27 February 2021 · Accepted: 19 April 2021 ·

Published Online: 27 May 2021



View Online



Export Citation



CrossMark

Chao Jiang (姜超)^{1,2,3}  Ricardo Vinuesa,⁴  Ruilin Chen (陈瑞林),^{1,2} Junyi Mi (米俊亦),^{1,2} Shujin Laima (赖马树金),^{1,2,3}  and Hui Li (李惠)^{1,2,3,a} 

AFFILIATIONS

¹Key Lab of Smart Prevention and Mitigation of Civil Engineering Disasters of the Ministry of Industry and Information Technology, Harbin Institute of Technology, Harbin 150090, China

²Key Lab of Structural Dynamics and Control of Ministry of Education, Harbin 150090, China

³Guangdong-Hong Kong-Macao Joint Laboratory for Data-driven Fluid Mechanics and Engineering Applications, Southern University of Science and Technology, Shenzhen 518055, China

⁴SimEx/Flow, Engineering Mechanics, KTH Royal Institute of Technology and Swedish e-Science Research Centre (SeRC), Stockholm 100 44, Sweden

^aAuthor to whom correspondence should be addressed: lihui@hit.edu.cn

ABSTRACT

Reynolds-averaged Navier–Stokes simulations represent a cost-effective option for practical engineering applications, but are facing ever-growing demands for more accurate turbulence models. Recently, emerging machine learning techniques have had a promising impact on turbulence modeling, but are still in their infancy regarding widespread industrial adoption. Toward their extensive uptake, this paper presents a universally interpretable machine learning (UIML) framework for turbulence modeling, which consists of two parallel machine learning-based modules to directly infer the structural and parametric representations of turbulence physics, respectively. At each phase of model development, data reflecting the evolution dynamics of turbulence and domain knowledge representing *prior* physical considerations are converted into modeling knowledge. The data- and knowledge-driven UIML is investigated with a deep residual network. The following three aspects are demonstrated in detail: (i) a compact input feature parameterizing a new turbulent timescale is introduced to prevent nonunique mappings between conventional input arguments and output Reynolds stress; (ii) a realizability limiter is developed to overcome the under-constrained state of modeled stress; and (iii) fairness and noise-insensitivity constraints are included in the training procedure. Consequently, an invariant, realizable, unbiased, and robust data-driven turbulence model is achieved. The influences of the training dataset size, activation function, and network hyperparameter on the performance are also investigated. The resulting model exhibits good generalization across two- and three-dimensional flows, and captures the effects of the Reynolds number and aspect ratio. Finally, the underlying rationale behind prediction is explored.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0048909>

I. INTRODUCTION

In flows of engineering interest, Reynolds-averaged Navier–Stokes (RANS) simulation is a cost-effective alternative to direct numerical simulation (DNS) and large eddy simulation (LES) in terms of grid resolution,¹ time steps,² and memory overhead.³ As discussed in recent reviews,^{4–6} there is an ever-growing demand for RANS adoption. However, there exist several limitations mainly due to inaccurate turbulence models in representing the real turbulence physics, as documented in the literature.^{6–9} Despite a significant need

for improved RANS models, conventional model development has almost stalled.

Recently, encouraging advances have been made in turbulence modeling using state-of-the-art machine learning (ML) algorithms.^{10–15} ML-augmented turbulence modeling provides a paradigm shift that can overcome the well-known bottleneck of conventional intuition-based model creations. Through this, the existing rich data from high-resolution simulations or experiments are utilized to inform closure models. The first attempt to apply an ML algorithm to

turbulence modeling goes back to Cheung *et al.*,¹⁶ who developed a Bayesian statistical model for the closure coefficients of the Spalart–Allmaras model. Since then, data-driven turbulence modeling has been advancing vigorously, e.g., Edeling *et al.*^{17,18} and Zhang and Fu,¹⁹ applying Bayesian inference to the coefficients of transport equations. Stochastic modeling always returns spatially independent coefficients. As an alternative, Weatheritt and Sandberg^{20–22} used gene expression programming (GEP) to explicitly recalibrate the closure coefficients of nonlinear algebraic models as functions of the local flow field, while Zhu *et al.*²³ reconstructed an algebraic surrogate to approximate the transport equation of the turbulent viscosity using a radial basis function. It is worth noting that Ling *et al.*²⁴ and Jiang *et al.*⁹ pioneered the use of deep neural networks (DNNs) to obtain spatially varying coefficients of nonlinear algebraic models. See also the latest similar studies.^{25–30} The aforementioned efforts are part of the ML-augmented parameter inference for a form-constrained model.

Rather than calibrating a set of coefficients to reduce the parametric uncertainty, adding source terms is another route to correct the existing models that focuses on improving the structural uncertainty. Dow and Wang³¹ made the first statistical attempt to estimate the discrepancy of the turbulent viscosity using a Gaussian process. To achieve the necessary dependence on the feature space, Duraisamy and co-workers^{32–34} proposed a promising data-driven framework comprised of Bayesian-based field inversion and ML-based function reconstruction, obtaining a mapping of the inferred discrepancy (e.g., a corrective multiplier or a direct corrective source) from local flow quantities. Following this practice, He *et al.*³⁵ adopted a continuous adjoint formulation to correct the Spalart–Allmaras model, and Yang and Xiao³⁶ used a DNN to correct the $k - \omega - \gamma - A_r$ transition model. Iaccarino and co-workers^{37–40} presented a novel approach to account for the structural uncertainty by directly perturbing the elements in the spectral decomposition of Reynolds stress. Although not an ML algorithm, this perturbation approach has achieved great success in engineering applications^{41–43} due to the following two prominent advantages. One is the good interpretability that perturbations injected into the turbulent kinetic energy, eigenvalues, and eigenvectors represent the modulations in the size, shape, and orientation of the Reynolds stress ellipsoid, respectively. Consequently, the ability to capture anisotropy is significantly improved, as recently unearthed by Mishra and Iaccarino.⁴⁴ More specifically, the perturbations to the turbulent kinetic energy accounts for the variability in the turbulent viscosity while the perturbations to eigenvalues and eigenvectors extend the isotropic eddy-viscosity model to an orthotropic and anisotropic version. Another one is its simple implementation of realizability (see Refs. 38 and 45). Xiao and co-workers^{46–48} benefited from the above two ideas and successfully used random forests to predict the discrepancy in the Reynolds stress projections. Yin *et al.*⁴⁹ followed this direction using a DNN. Notably, the philosophy behind adding source terms is, although not necessarily reasonable, that the existing structural terms are able to reproduce the dominant part of turbulence physics.

In comparison, the ultimate barrier in turbulence modeling is to develop a universal model across multiple classes of flows. This requires a more flexible predictive framework to reduce both the structural and parametric uncertainties, aiming at developing new models free from the existing ones. Early attempts have been made to directly predict the targeted stress using ML algorithms^{50–55} (see Refs. 56 and

57 for direct modeling of turbulent scalars). However, as stressed by Edeling *et al.*,⁵⁸ the structural form of the predicted stress is unknown. As an alternative, ML-based sparse regression approaches^{59–61} have been successfully applied to distill the underlying exact relations behind data, with some initial attempts in discovering the governing equations for fluid dynamics.^{62–65} Despite a promising approach, the data-driven discovery for RANS models could be considerably difficult because there is no *prior* knowledge about the candidate structural terms. Even if the exact closure models are eventually discovered in terms of partial differential equations, they are still computationally expensive. Thus, it is practical to obtain optimal models in a user-defined sense.

The current status therefore motivates us to design two parallel ML-based modules to directly infer both the structural bases and closure coefficients based on data and domain knowledge, thus achieving the structural and parametric representations of turbulence physics, respectively. The two modules interact through a multiplicative layer, analogous to tensor modeling. Each learned structural basis and corresponding referred closure coefficient make up one term of the overall functional model form. It is well believed that different flows can be simulated properly using the same form-constrained model with different coefficients, indicating that the closure coefficients reflect the physical effects of the flow boundary. As a result, the proposed framework may serve as a transfer learning framework for dynamic tasks: one can retrain the data-driven model by updating only the closure coefficients, with fixed learned structural bases (as known modeling knowledge). This would be an advantage over previous models using a single ML-based module to directly predict the Reynolds stress. Besides, a DNN is selected as the ML algorithm for our framework due to its clear advantages over random forests in extracting features from informative data,⁶⁶ capturing invariances,⁶⁷ and the approximation capability.⁵¹

To investigate the performance of the proposed framework by itself, a controlled environment will be needed in which the learning dynamics of the framework, e.g., sources of gains and the upper-performance limit, can be explored in isolation. Thus, we first constrain the structural bases to the conventional tensor bases of Pope⁶⁸ as a proof-of-concept. Against this backdrop, we perform a systematic investigation in three aspects: (i) how to select input features and utilize *prior* domain knowledge; (ii) how to set up a fair and robust learning strategy to overcome prediction prejudice and noise-sensitivity; and (iii) how to evaluate a data-driven model reasonably. Note that previous approaches of Ling *et al.*²⁴ and Beetham and Capecelatro²⁷ can be regarded as special cases of our proposed model.

Overall, this work seeks to develop a universally interpretable machine learning (UIML) framework of turbulence modeling with built-in fundamental principles, under which a physics-informed residual network (PiResNet) is achieved. The rest of this paper is structured as follows. Section II introduces fundamental aspects of UIML, focusing especially on integrating structural domain knowledge into PiResNet, e.g., feature selection as inputs, physical constraints of outputs, and a corrected turbulent timescale. Section III covers a fair and robust training strategy for PiResNet, aiming at achieving high-fidelity predictions with unbiased generalization and lower noise-sensitivity. Section IV concerns the predictive performance of PiResNet, including a *prior* assessment by examining its accuracy, realizability, and robustness, and a *posterior* validation by inserting PiResNet into the

governing dynamical systems to directly predict the mean-flow responses for different flow configurations. Section V further explores the underlying rationale behind prediction, and the potentials and limitations. Finally, Sec. VI draws the main findings and highlights some remaining key challenges for future extensions of this work.

II. METHODOLOGY

The incompressible RANS equations for the resolved flow variables can be stated as

$$\nabla \cdot \mathbf{U} = 0, \quad (1a)$$

$$\partial_t \mathbf{U} + \mathbf{U} \cdot \nabla \mathbf{U} + \nabla p - \nu \nabla^2 \mathbf{U} = -2 \nabla \cdot k \mathbf{a}, \quad (1b)$$

where \mathbf{U} , p , k , and ν are the mean-flow velocity vector, nominal pressure (including the kinematic pressure and isotropic part of Reynolds stress), turbulent kinetic energy, and kinematic viscosity, respectively. The stress anisotropy tensor, $\mathbf{a} \equiv \tau/(2k) - \mathbf{I}/3$ (where τ is the Reynolds stress tensor and \mathbf{I} is the identity tensor), represents the effects of unresolved turbulence. For the purpose of closure, the stress anisotropy tensor can only be related to the mean-flow field.

A. Summary of the proposed UIML framework

Turbulence closure modeling, at the RANS level, is a typical multivariate problem regarding the stress anisotropy tensor. Tensor modeling is well known as a good physical option in a multivariate problem due to its objectivity. Generally, a given multivariate problem can be formulated to a polynomial of a set of structural bases with corresponding closure coefficients, as described by Lumley.⁶⁹ Accordingly, we design two parallel ML-based modules, as depicted in Fig. 1, for distilling the structural bases and closure coefficients from data, both of which are integrated through a multiplicative layer to constitute the stress anisotropy tensor. In either module, there is enough flexibility to build the desired DNN architecture, e.g., plain neural network,⁷⁰ residual neural network,⁷¹ or Bayesian neural network,⁷² thus indicating a universal, inherently interpretable ML-based framework for turbulence modeling. According to the concept defined by Rudin,⁷³ the interpretability of the resulting UIML framework lies, first and foremost, in its constrained model form as

$$f_1: \mathbf{q} \mapsto \mathbf{g}, \quad f_2: \mathcal{Q} \mapsto \mathcal{T}, \quad \mathbf{a} = \mathbf{g} \cdot \mathcal{T}, \quad (2)$$

where $\mathbf{q} = \{q_1, q_2, \dots\}$ and $\mathcal{Q} = \{\mathbf{Q}^{(1)}, \mathbf{Q}^{(2)}, \dots\}$ are two vectors of scalar-valued input features q_i and tensor-valued input features $\mathbf{Q}^{(i)}$; $\mathbf{g} = \{g_1, g_2, \dots\}$ and $\mathcal{T} = \{\mathbf{T}^{(1)}, \mathbf{T}^{(2)}, \dots\}$ are two vectors of learned scalar-valued closure coefficients g_i and tensor-valued structural bases $\mathbf{T}^{(i)}$. The resulting model represented by Eq. (2) has the structure $\mathbf{a} = \sum g_i \mathbf{T}^{(i)}$, which helps us to interpret what the UIML has learned from informative data. f_1 and f_2 account for parametric and structural representations of turbulence physics, respectively. The modeling process is straightforward. The transparency of the UIML obtains the superiority of symbolic regression based on evolution algorithms.^{20,21} In contrast, Edeling *et al.*³⁸ stressed that there is no analytic form for interpreting the structure of learned model discrepancies by other data-driven approaches.^{32,36,47} In particular, when the structural bases $\{\mathbf{T}^{(i)}\}_{i=1}^{10}$ are constrained to those of Pope,⁶⁸ the UIML degrades to a general form of conventional algebraic models. Thus, the UIML is a more universal framework than existing models based on representation theorems (e.g., Hilbert basis theorem⁷⁴ and Cayley–Hamilton theorem⁷⁵) and has more flexibility. Second, the UIML obeys *prior* domain knowledge that we inject into it hereafter, such as invariant properties (rotational, reflectional, extended Galilean, and scale invariances), corrective turbulent timescale, realizability constraints, unbiased principles, and robust regularizations. Alber *et al.*⁷⁶ conducted a systematic investigation of ML-based multiscale modeling with no fundamental laws of physics and encountered unphysical solutions and ill-posed problems. Chang and Dinh¹¹ and Pawar *et al.*⁷⁷ also stressed the significance of domain knowledge. All *prior* domain knowledge is thus reasonably converted into modeling knowledge for the UIML during our model development. Third, the unbiased and robust learning strategy adopted in this work helps interpret how the UIML achieves fair and noise-insensitive predictions. Developing an inherently interpretable model is of critical importance for physical problems. However, interpretability has not been reasonably considered. The UIML is an attempt to bridge this gap.

After designing the UIML, training data are utilized to establish a DNN-based surrogate of \mathbf{a} by minimizing the following cost function and determining the optimal trainable parameters θ_π

$$\theta_\pi = \underset{\theta}{\operatorname{argmin}} \{ \| \tilde{\mathbf{a}} - \mathbf{g} \cdot \mathcal{T} \|_F^2 + \lambda_g \| \mathbf{g} \|_1 + \lambda_w \| \mathbf{w} \|_2^2 \}, \quad (3)$$

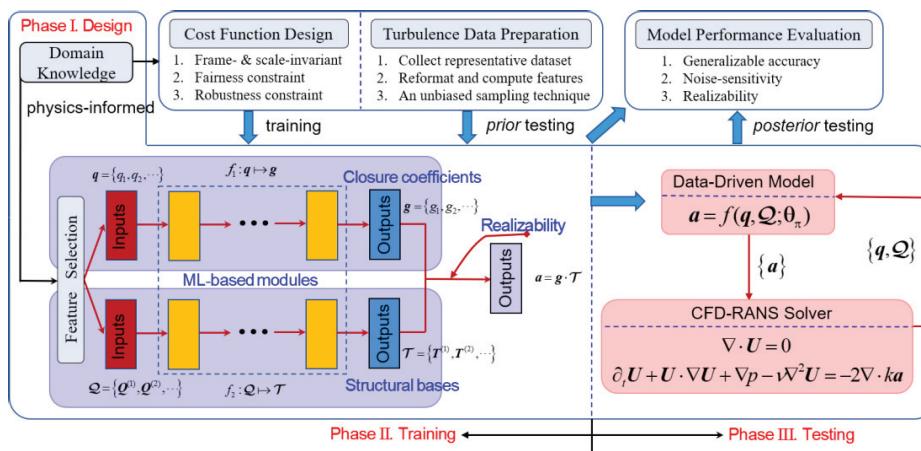


FIG. 1. Development lifecycle of the universally interpretable machine learning (UIML) framework for turbulence modeling containing three phases: design of the UIML with domain knowledge (Phase I), a fair and robust training strategy for the UIML (Phase II), and systematic performance testing of the UIML (Phase III).

where $\|\cdot\|_F$, $\|\cdot\|_1$, and $\|\cdot\|_2$ denote the Frobenius norm, L_1 norm, and L_2 norm, respectively, and $\tilde{\alpha}$ denotes the true labels. The three terms in Eq. (3) are the training error, a penalty applied to the closure coefficients for sparsity, and a penalty applied to the trainable parameters for robustness, respectively. λ_g and λ_w are penalty factors (see also Sec. III C). The resulting data-driven model reads

$$\boldsymbol{a} = f(\boldsymbol{q}, \mathcal{Q}; \boldsymbol{\theta}_\pi). \quad (4)$$

Finally, we insert Eq. (4) into the computational fluid dynamics (CFD) environment represented by Eqs. (1a) and (1b) and evaluate its predictive skills for the quantities of interest (QoIs) in unseen flows.

The overall workflow of the UIML is summarized in Fig. 1. A physically motivated selection guideline of input features (e.g., mean-flow field variables, invariance properties, and compactness) and the corrective turbulent timescale considering the viscous effects are adopted in the design phase. A frame- and scale-invariant cost function as well as regularization constraints on both the trainable parameters and closure coefficients is implemented in the training phase. Realizability properties to constrain the learned model are enforced both in the training and testing phases. Through the fusion of informative data and domain knowledge, the developed model under the UIML is both data- and knowledge-driven.

B. Physical requirements of stress anisotropy

All reasonable principles regarding conventional turbulence modeling are strictly respected in data-driven turbulence modeling under the UIML. Most importantly, given the rotational and reflectional invariances⁷⁸ of the RANS equations represented by Eq. (1), the overall regression function f representing the data-driven model must be form-invariant with respect to rotations and reflections of the coordinate system. That is, a transformation of f under an arbitrary orthogonal tensor \mathbf{R} ($\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}$ and $\det \mathbf{R} = \pm 1$) should satisfy the relation,

$$f(\boldsymbol{q}, \mathcal{R}\mathcal{Q}\mathbf{R}^T; \boldsymbol{\theta}_\pi) = \mathbf{R}[f(\boldsymbol{q}, \mathcal{Q}; \boldsymbol{\theta}_\pi)]\mathbf{R}^T. \quad (5)$$

The superscript T denotes the tensor transpose. Note that RANS equations are invariant under rotational and reflectional transformations of the coordinate system, while none of the terms in RANS equations (including the stress anisotropy tensor) are rotational or reflectional invariants. Accordingly, we seek to ensure rotational and reflectional invariances of the regression function f , rather than the stress anisotropy tensor itself. Obviously, the scalar-valued function $f_1 : \boldsymbol{q} \mapsto g$ as a function of scalars is invariant with respect to rotations and reflections of the coordinate system. Thus, Eq. (5) reduces to

$$\mathbf{R}\mathbf{T}\mathbf{R}^T \equiv \mathbf{R}f_2(\mathcal{Q})\mathbf{R}^T = f_2(\mathcal{R}\mathcal{Q}\mathbf{R}^T). \quad (6)$$

To satisfy Eq. (6), $f_2 : \mathcal{Q} \mapsto \mathcal{T}$ should be designed as a tensor-valued isotropic function of the arguments. Speziale *et al.*⁷⁹ and Pope⁶⁸ explicitly constructed such functions for pressure-strain correlation and the stress anisotropy tensor, respectively. In comparison, Eq. (6) is represented by an ML-based module. However, it is often misunderstood that the input-output equivariance can be guaranteed by choosing rotational and reflectional invariants as input features (e.g., see Refs. 39 and 40). Considering that there is no explicit analytic form (as an aside, although analytic forms strictly exist, they are considerably

complex to understand) in most data-driven turbulence modeling, one must take care with the construction of pre-defined functions by the adopted ML algorithm. Ling *et al.*⁶⁷ used data augmentation to preserve approximate invariances under rotational and reflectional transformations. Furthermore, regarding the dynamical processes of the RANS equations, the Reynolds stress τ reduces from Euclidean invariance to extended Galilean invariance,⁸⁰ not just Galilean invariance as is usually stated. It is therefore easily deduced that the stress anisotropy \boldsymbol{a} also has extended Galilean invariance owing to the extended Galilean invariance of both the turbulent kinetic energy k and the Reynolds stress τ . When using isotropic functions, the extended Galilean invariance of the predicted stress anisotropy \boldsymbol{a} can be easily guaranteed by constraining the extended Galilean invariance to the input features $\{\boldsymbol{q}, \mathcal{Q}\}$. However, the input features in most previous studies do not have extended Galilean invariance (some are not even Galilean invariants), which could destroy the properties of the predicted Reynolds stress and the extrapolation capability. Additionally, dimensionless stress anisotropy is scale-invariant owing to the Reynolds-number similarity of the RANS equations. Dimensionless input features are therefore needed. Last but not least, the stress anisotropy is bounded by a set of inequality constraints, i.e., realizability requirements, which have been comprehensively studied by Lumley⁸¹ and Banerjee *et al.*⁸² However, these constraints are not always satisfied in previous models. All four of the above-mentioned considerations are important prerequisites for modeling the stress anisotropy and are discussed below.

C. General principle for selecting input features

The present work seeks a deterministic model that depends on the feature space, rather than a spatially independent stochastic model. Thus, we need to construct proper input features from the mean-flow field $\{\nabla \mathbf{U}, \nabla p, k, \varepsilon, \nu\}$, where ε is the turbulent dissipation rate of k . The fundamental principle for selecting the input features $\{\boldsymbol{q}, \mathcal{Q}\}$ is according to the requirements of the targeted stress anisotropy \boldsymbol{a} and the properties of the regression function f . Notably, \boldsymbol{a} is selected as the predicted target instead of the Reynolds stress τ , as two dynamically similar flows share the same \boldsymbol{a} but have different τ . Thus, one can benefit from two aspects when using \boldsymbol{a} : (i) more compact training data, thus reducing the computational cost, and (ii) a scale-invariant cost function that guarantees the validity of the hyperparameters in the DNN across dynamically similar flows.

As mentioned above, \boldsymbol{a} has four physical properties: (i) extended Galilean invariance, (ii) scale invariance, (iii) symmetry, and (iv) realizability. When f_1 and f_2 are isotropic functions, the input features must obey the same principles as \boldsymbol{a} , except for the last one. The realizability constraints are a set of inequalities among components of \boldsymbol{a} and are thus directly enforced to the predicted \boldsymbol{a} . In addition, the selected input features must have clear physical justifications and be compact. With more input features, the learned model becomes more complex and more difficult to understand.

The mean strain-rate and rotation-rate tensors $\mathbf{S} \equiv (\nabla \mathbf{U} + \nabla \mathbf{U}^T)/2$ and $\mathbf{Q} \equiv (\nabla \mathbf{U} - \nabla \mathbf{U}^T)/2$ are widely used field variables in flow analysis and conventional turbulence modeling. More recently, their significant influences on the evolution dynamics of turbulence, in accordance with the conventional intuition, have been confirmed by Jiménez.⁸³ In the two test flows (see Sec. III A), only ∇p is excluded,

thus leading to a general mapping from the mean-flow field to the stress anisotropy tensor

$$\boldsymbol{a} = \boldsymbol{a}(\mathbf{s}, \boldsymbol{\Omega}, k, \varepsilon, \nu). \quad (7)$$

Obviously, all input arguments have extended Galilean invariance. Notably, there are two things to be clarified. One is that the extended Galilean invariance of the input features has rarely been considered by a reasonable means to reproduce the extended Galilean invariance of \boldsymbol{a} . Input features that are only Galilean invariants could destroy the physical properties of \boldsymbol{a} , i.e., qualitatively incorrect. The second point to clarify is that, as recently demonstrated by Huang and Durst⁸⁰ for a dynamical process, \boldsymbol{a} is not frame-indifferent. This implies that $\boldsymbol{\Omega}$ can serve as an effective constitutive argument in turbulence modeling which is different from the guiding principle of continuum mechanics put forth by Noll.⁸⁴

To infer the structural bases, i.e., $f_2 : \mathcal{Q} \mapsto \mathcal{T}$, the input features and outputs should be dimensionless, and have extended Galilean invariance. The outputs should also be symmetric. Thus, $\mathbf{s} \equiv k\mathbf{s}/\varepsilon$ and $\boldsymbol{\omega} \equiv k\boldsymbol{\Omega}/\varepsilon$ are selected as the input features. When constraining $\mathcal{T} = \{T^{(i)}\}_{i=1}^{10}$ to the tensor bases given by Pope,⁶⁸ a special version of the UIML is obtained as

$$\begin{aligned} \boldsymbol{a} = & g_1 \mathbf{s} + g_2 \left[\mathbf{s}^2 - \frac{1}{3} \text{tr}(\mathbf{s}^2) \mathbf{I} \right] + g_3 (\boldsymbol{\omega} \mathbf{s} - \mathbf{s} \boldsymbol{\omega}) \\ & + g_4 \left[\boldsymbol{\omega}^2 - \frac{1}{3} \text{tr}(\boldsymbol{\omega}^2) \mathbf{I} \right] + \dots, \end{aligned} \quad (8)$$

where $\text{tr}(\cdot)$ denotes the trace. To infer the closure coefficients, i.e., $f_1 : \mathbf{q} \mapsto \mathbf{g}$, traditional flow variables (related to the traces of structural bases) are selected as input features. For instance, $s_m \equiv (2s_{kl}s_{kl})^{1/2}$ and $\omega_m \equiv (2\omega_{kl}\omega_{kl})^{1/2}$ have extended Galilean invariance and scale invariance, and are related to the positive second invariant of $\nabla \mathbf{U}$, i.e., $Q_c \equiv (\omega_m^2 - s_m^2)/4$, which is a criterion⁸⁵ for detecting vortical structures. Note that Eq. (8) cannot achieve satisfactory predictions under conventional tensor invariants $\mathbf{q} = \{s_m, \omega_m, \dots\}$. The reasons for the failure of Eq. (8) are analyzed in Sec. II D. Ling *et al.*²⁴ and Weatheritt and Sandberg²¹ encountered this difficulty, which was also noted recently by Geneva and Zabaras.⁸⁶ For compactness, the present work provides a solution with only one additional input feature, as shown below.

D. Preserving uniqueness with a new timescale

To establish a deterministic ML-based predictor (i.e., a data-driven model), a unique mapping from the selected input features to the targeted outputs in the training dataset is a fundamental prerequisite. However, using $\mathbf{q} = \{s_m, \omega_m, \dots\}$ and $\mathcal{Q} = \{\mathbf{s}, \boldsymbol{\omega}\}$ as input

features cannot satisfy this condition in turbulent flows. We apply Eq. (8) to two-dimensional fully developed channel flows to demonstrate the drawbacks and analyze the causes. In such flows, only dU/dy (U is the mean velocity magnitude and y is the wall-normal direction) exists. Accordingly, all nonzero mean-flow quantities are as follows:

$$s_{12} = s_{21} = \frac{1}{2} s_m, \quad \omega_{12} = -\omega_{21} = \frac{1}{2} \omega_m, \quad s_m = \omega_m = \frac{k dU}{\varepsilon dy}. \quad (9)$$

The flow is controlled by the dominant shear stress a_{12} . Substituting Eq. (9) into Eq. (8) leads to

$$a_{12} = \frac{1}{2} g_1 s_m - \frac{1}{4} g_6 s_m \omega_m^2, \quad (10)$$

where the coefficients g_1 and g_6 are ultimately dependent on s_m . Thus, Eq. (10) can be expressed as a univariate function of s_m . Figure 2(a) shows the true relationship between a_{12} and s_m at $Re_\tau = 5200$,⁸⁷ exhibiting unexpected nonunique mappings, i.e., $s_m(y_A^+) = s_m(y_B^+)$, and $a_{12}(s_m)|_{y_A^+} \neq a_{12}(s_m)|_{y_B^+}$. Obviously, Eq. (10) is inconsistent with Fig. 2(a). As a result, Eq. (8) cannot be used as the regression function to establish a functional mapping from the selected s_m to the targeted a_{12} for a given training dataset.

The data-driven models of Ling *et al.*²⁴ (denoted as TBNN) and Weatheritt and Sandberg²¹ can also be represented by Eq. (8). We take TBNN as an example to quantitatively illustrate the drawbacks of Eq. (8). Consider two training scenarios: TBNN-1 is trained on the whole dataset while TBNN-2 is trained using data far from the wall (inner-scaled $y^+ > 9$). Both results are shown in Fig. 2(b). TBNN-1 displays worse training performance than TBNN-2 (nonunique mappings only in the circular region, see Fig. 1), indicating that nonunique mappings in the training dataset make training more difficult. Recently, Geneva and Zabaras⁸⁶ also pointed out that TBNN is difficult to train and yields unsatisfactory predictions, but did not offer a specific solution. Actually, this problem is common for models based on Eq. (8).

From a mathematical standpoint, there are too few input features. The relations $\mathbf{s}(y_A^+) = \mathbf{s}(y_B^+)$ and $\boldsymbol{\omega}(y_A^+) = \boldsymbol{\omega}(y_B^+)$ result in the same values for the structural bases at locations A and B, i.e., $T^{(i)}(y_A^+) = T^{(i)}(y_B^+)$ ($i = 1, 2, \dots, 10$). To return different values for \boldsymbol{a} at locations A and B as expected in Fig. 2(a), $g_i(y_A^+) \neq g_i(y_B^+)$ must be guaranteed. Thus, an additional input feature q_3 should be introduced, which must be extended Galilean invariant and dimensionless, and satisfy the following relation under any $s_m(y_A^+) = s_m(y_B^+)$:

$$q_3(y_A^+) \neq q_3(y_B^+). \quad (11)$$

From a physical standpoint, an inappropriate turbulent timescale is chosen in Eq. (8). There are two assumptions in deriving Eq. (8)

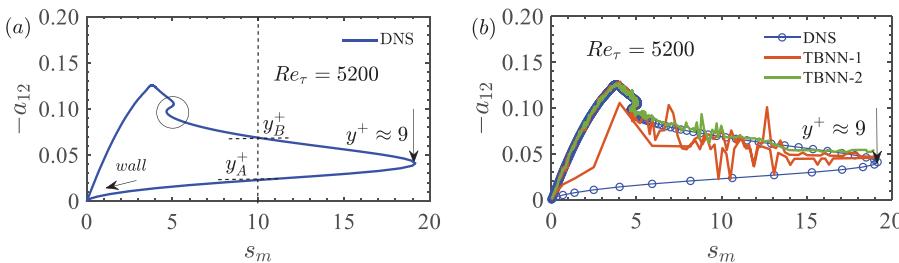


FIG. 2. Two-dimensional developed channel flow at $Re_\tau = 5200$: (a) shear stress anisotropy a_{12} against characteristic strain rate s_m from DNS,⁸⁷ and (b) two TBNN-predicted a_{12} against s_m with different meshes.

from Eq. (7): (i) the feature ν is excluded as an argument, and (ii) the macro-timescale of turbulence k/ε is chosen as the flow timescale from the wall to the fully turbulent region. Lumley⁶⁹ showed that Reynolds stresses are uniquely related to the mean-flow field and macro-timescale of turbulence for a nearly homogeneous flow. Hence, conventional turbulence models only offer good predictions in regions far from the wall. Jiang *et al.*⁹ conducted a comparative study to demonstrate this issue. In the near-wall region, a correction is needed to account for the dominant viscous effect. According to the expansions of the turbulent field near the wall by Hanjalic and Launder,⁸⁸ the turbulent timescale (i.e., the ratio of the length scale of the energy-containing eddies to the turbulent velocity scale) approaches a nonzero value. However, the commonly used timescale k/ε vanishes at the wall because $k \rightarrow 0$. The timescale is expected to be the Kolmogorov timescale $\sqrt{\nu/\varepsilon}$ because viscous dissipation dominates near the wall. Thus, a new turbulent timescale is defined as

$$\tau_t \equiv \sqrt{(k/\varepsilon)^2 + c_t^2(\nu/\varepsilon)} = \sqrt{\lambda}(k/\varepsilon), \quad \lambda \equiv 1 + c_t^2/Re_t, \quad (12)$$

where $c_t (> 0)$ is a weighting parameter and $Re_t \equiv k^2/(\nu\varepsilon)$ is the turbulent Reynolds number that is widely used in low-Reynolds-number models.^{89–92} $\lambda (\geq 1)$ reflects the ratio of the new timescale to the conventional timescale. The new timescale τ_t has good properties: (i) it recovers to the standard k/ε at large Re_t , and (ii) it never falls below $c_t\sqrt{\nu/\varepsilon} (Re_t \ll 1)$. It is worth noting that τ_t can prevent the singularity at the wall that results from k/ε vanishing.

Using $s' \equiv \tau_t \mathbf{s}$ and $\omega' \equiv \tau_t \boldsymbol{\Omega}$, Eq. (8) can be rewritten as

$$\begin{aligned} \mathbf{a} &= g' \mathbf{s}' + g'_2 \left[\mathbf{s}'^2 - \frac{1}{3} \text{tr}(\mathbf{s}'^2) \mathbf{I} \right] + g'_3 (\mathbf{s}' \boldsymbol{\omega}' - \boldsymbol{\omega}' \mathbf{s}') \\ &\quad + g'_4 \left[\boldsymbol{\omega}'^2 - \frac{1}{3} \text{tr}(\boldsymbol{\omega}'^2) \mathbf{I} \right] + \dots, \end{aligned} \quad (13)$$

with all closure coefficients determined using $f_1 : \mathbf{q}' \mapsto \mathbf{g}'$, where $\mathbf{q}' = \{s'_m, \omega'_m, \dots\}$, $s'_m = \sqrt{\lambda}s_m$, and $\omega'_m = \sqrt{\lambda}\omega_m$. Thus, in two-dimensional fully developed channel flows, a_{12} of Eq. (13) eventually depends on both s_m ($s_m = \omega_m$) and Re_t at a given c_t . In the following, we examine whether Re_t confronts the nonunique mapping problem. As Fig. 3(a) shows, at any two locations with the same strain, i.e., $s_m(y_A^+) = s_m(y_B^+)$, there always exist two different values of Re_t , i.e., $Re_t(y_A^+) \neq Re_t(y_B^+)$. Similarly, at any two locations with $Re_t(y_A^+) = Re_t(y_B^+)$, there always exist $s_m(y_A^+) \neq s_m(y_B^+)$. That is, $s_m(y_A^+) = s_m(y_B^+)$ and $Re_t(y_A^+) = Re_t(y_B^+)$ do not occur at the same time. The selection of Re_t satisfies the condition of Eq. (11) and successfully

overcomes the nonunique input-output mappings in conventional models represented by Eq. (8).

The final question is how to determine the value of c_t in order to directly utilize Eq. (13). We note that $\lambda > 1$ implies a correction to the conventional timescale k/ε . Thus, a different value of c_t produces a different range where the correction is activated. In the viscous sublayer ($y^+ < 5$), $k = cy^2$, where c is inversely proportional to the square of the timescale.⁹¹ We set $c = 1/(c_t^2\nu/\varepsilon)$ in the vicinity of the wall. The transport equation of k reduces to $\nu\partial_k y^2 = \varepsilon$ at the wall.⁹³ Combining the above relations yields $c_t = \sqrt{2}$. This derivation implies that a correction takes place only in the viscous sublayer. The variation of λ at different values of c_t is further investigated based on DNS in Fig. 3(b). Obviously, $c_t = \sqrt{2}(n=1)$ corresponds to a correction within $y^+ < 10$ (slight departure from $y^+ < 5$). Similarly, $c_t = 4(n=4)$ and $c_t = 8\sqrt{2}(n=7)$ correspond to corrections within $y^+ < 50$ and $y^+ < 500$, respectively. However, there is no *prior* knowledge about the exact upper bound below which a correction is applied. Consequently, a strictly reasonable value of c_t may not be available. Thus, Eq. (13) cannot be directly used for turbulence modeling because the input features $\mathbf{q}' = \{s'_m, \omega'_m, \dots\}$ and $\mathbf{Q}' = \{s', \omega'\}$ are undetermined. To overcome this difficulty, we re-form Eq. (13) as

$$\begin{aligned} \mathbf{a} &= c_1 \mathbf{s} + c_2 \left[\mathbf{s}^2 - \frac{1}{3} \text{tr}(\mathbf{s}^2) \mathbf{I} \right] + c_3 (\mathbf{s}\boldsymbol{\omega} - \boldsymbol{\omega}\mathbf{s}) \\ &\quad + c_4 \left[\boldsymbol{\omega}^2 - \frac{1}{3} \text{tr}(\boldsymbol{\omega}^2) \mathbf{I} \right] + \dots, \end{aligned} \quad (14)$$

where the closure coefficients $c_1 \equiv \sqrt{\lambda}g'_1$, $c_2 \equiv \lambda g'_2$, $c_3 \equiv \lambda g'_3$, and $c_4 \equiv \lambda g'_4$ can eventually be expressed as functions of $\{s_m, \omega_m, \dots\}$ and Re_t if c_t is known. The advantage of Eq. (14) is that the dependence of Eq. (13) on c_t is entirely imputed to the closure coefficients $\{c_i\}_{i=1}^{10}$. The unknown parameter c_t can be further implicitly considered by the ML-based modules when using $\{s_m, \omega_m, \dots\}$ and Re_t as input features to directly regress $\{c_i\}_{i=1}^{10}$. The problem is now closed. Notably, although Eq. (14) is directly used for data-driven turbulence modeling, its physical meaning lies in Eq. (13), which interprets why Re_t should be included as an input feature.

E. Enforcing realizability constraints to outputs

As noted previously, realizability is a physically valuable guiding principle in developing turbulence models which is useful in ensuring numerical stability. Efforts have been made to achieve realizable solutions in conventional models,^{94–96} but few studies have been

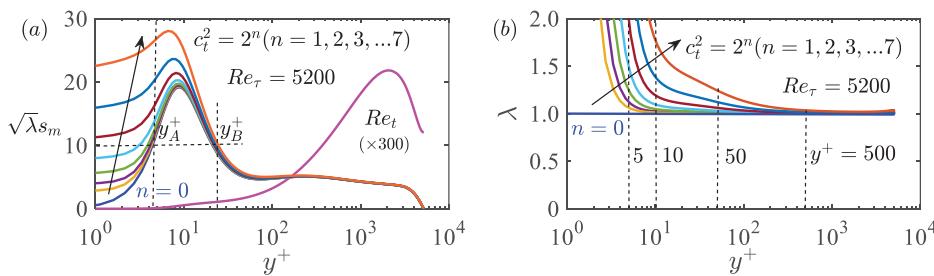


FIG. 3. Two-dimensional fully developed channel flow at $Re_t = 5200$:⁸⁷ (a) shear parameter s_m and turbulent Reynolds number Re_t along the wall-normal direction y^+ , and (b) ratio of new timescale to macro-timescale along the wall-normal direction y^+ . The results at different values of the weighting parameter c_t are shown.

considered in data-driven turbulence modeling. We directly enforce realizability constraints to the targeted stress anisotropy, rather than to the closure coefficients themselves. An implementation for the closure coefficients may be practicable in conventional models, but is considerably difficult in data-driven models because the closure coefficients are “implicitly” represented by an ML-based module.

Schumann⁹⁷ showed that realizability constraints contain a set of inequalities; however, it is difficult to use these inequalities as an effective correction strategy in computational codes. Instead, we select the following inequalities:

$$a_{\beta\beta} \geq -\frac{1}{3}, \quad (15a)$$

$$a_{\beta\gamma}^2 \leq \left(a_{\beta\beta} + \frac{1}{3} \right) \left(a_{\gamma\gamma} + \frac{1}{3} \right), \quad (15b)$$

$$\lambda_1 \geq \frac{3|\lambda_2| - \lambda_2}{2}, \quad (15c)$$

$$\lambda_1 \leq \frac{1}{3} - \lambda_2, \quad (15d)$$

where $\lambda_i (i = 1, 2, 3)$ are the eigenvalues of the stress anisotropy \mathbf{a} ($\lambda_1 \geq \lambda_2 \geq \lambda_3$). Note that no summations are taken for Greek indices. The above conditions are mathematically rigorous and physically important. The first condition, Eq. (15a), prevents negative energy. The second condition, Eq. (15b), originates from the Schwarz inequality and offers the upper bound for the shear stresses. The remaining conditions, Eqs. (15c) and (15d), ensure that the Reynolds stress is positive semidefinite. Thus, Eqs. (15a)–(15d) are sufficient for satisfying the realizability constraints.

Despite selecting constrained conditions, a correction strategy should be further developed when Eqs. (15a)–(15d) are not completely satisfied. Accordingly, we propose a progressive iteration realizability (PIR) scheme as the corrector (see Algorithm 1). The PIR scheme is applied to both training and testing phases, thus achieving a physically consistent and numerically stable data-driven model. In comparison, the conditions adopted by Ling *et al.*²⁴ are under-constrained. Comparative testing is conducted in Sec. IV B.

III. TRAINING: DATA PREPARATION AND LEARNING STRATEGY

In this section, we present a systematic ML procedure for training a data-driven turbulence model. In particular, we focus on how to design a fair and robust learning strategy in order to achieve unbiased and noise-insensitive predictions.

Fairness in data-driven turbulence modeling is a completely new concept, although it has long been an important consideration in high-risk decision-making scenarios. As a domain-specific concept, fairness herein refers to the *absence of any prejudice or favoritism in predicted outcomes toward a specific flow domain, stress component, and coordinate system*. Thus, unfairness could arise from hidden or neglected biases in the selection of training data or cost function. We present two fairness measures: a clustering-based data sampling technique and a fair cost function design.

ALGORITHM 1: Progressive Iteration Realizability (PIR) Scheme.

Inputs: Stress anisotropy tensor \mathbf{a} .

```

 $\mathbf{a} \leftarrow (\mathbf{a} + \mathbf{a}^T)/2;$ 
for  $N$  iterations do
  while  $\min\{a_{\beta\beta}\} < -1/3$  do
     $a_{\beta\beta} \leftarrow -a_{\beta\beta}[\min\{a_{\beta\beta}\}]^{-1}/3$ ; // narrow nonzero trace
  end
  while  $a_{\beta\gamma}^2 > (a_{\beta\beta} + 1/3)(a_{\gamma\gamma} + 1/3)$  do
     $a_{\beta\gamma} \leftarrow \sqrt{\max\{(a_{\beta\beta} + 1/3)(a_{\gamma\gamma} + 1/3), 0\}} \cdot \text{sign}\{a_{\beta\gamma}\}$ ;
  end
```

Compute eigenvalues of \mathbf{a} : $\lambda_1 \geq \lambda_2 \geq \lambda_3$;

```

while  $\lambda_1 < (3|\lambda_2| - \lambda_2)/2$  do
   $\lambda_j \leftarrow \lambda_j(3|\lambda_2| - \lambda_2)/2\lambda_1$ ; // as an amplifier
end
while  $\lambda_1 > 1/3 - \lambda_2$  do
   $\lambda_j \leftarrow \lambda_j(1/3 - \lambda_2)/\lambda_1$ ; // as a reducer
end
```

Reconstruct \mathbf{a}' with updated eigenvalues and initial eigenvectors;

```

 $a_{\beta\gamma} \leftarrow (a'_{\beta\gamma} + a'_{\gamma\beta})/2 (\beta \neq \gamma)$ ; // only update off-diags to retain zero trace
end
```

Outputs: Return well-constrained \mathbf{a} .

Another concern is the robustness of a data-driven model. Robustness herein refers to the *insensitivity of a data-driven model to a certain degree of noise or perturbation in the input features*. Taking the model of Gatski and Speziale⁹⁸ as an example, the turbulent viscosity can be expressed as

$$\frac{3}{3 - 2c_\eta^2 + c_\xi^2} \approx \frac{3(1 + c_\eta^2)}{3 + c_\eta^2 + 6c_\xi^2 c_\eta^2 + 6c_\xi^2} \approx \frac{1}{1 + 4c_\eta^2 + c_\xi^2}, \quad (16)$$

where c_ξ and c_η are strain-dependent coefficients. The right-hand side⁹⁹ of Eq. (16) exhibits better numerical stability than the other terms because the resulting turbulent viscosity is relatively insensitive to small perturbations in the strain. Jiang *et al.*⁹ summarized the general principle for conventional explicit models. In comparison, there have been no such considerations in “implicit” data-driven turbulence modeling. Our work is an initial attempt.

A. Representative dataset for training and testing

Representative datasets of turbulent flows should be carefully selected to investigate the predictive skills of a data-driven turbulence model in capturing different turbulent phenomena. Two-dimensional fully developed channel flow is a general representation of wall-bounded flows. We use this simple parallel-shear flow to demonstrate whether an additional input feature, Re_t , can capture those flow characteristics near the wall that cannot be adequately predicted by most conventional models. Four DNS datasets at various friction Reynolds numbers (based on the friction velocity u_τ and half-channel height h) are used: (i) $Re_\tau = 650$,¹⁰⁰ (ii) $Re_\tau = 1000$,¹⁰¹ (iii) $Re_\tau = 2000$,¹⁰² and (iv) $Re_\tau = 5200$.⁸⁷ A three-dimensional fully developed duct flow is selected as the second canonical example due to its rich characteristics

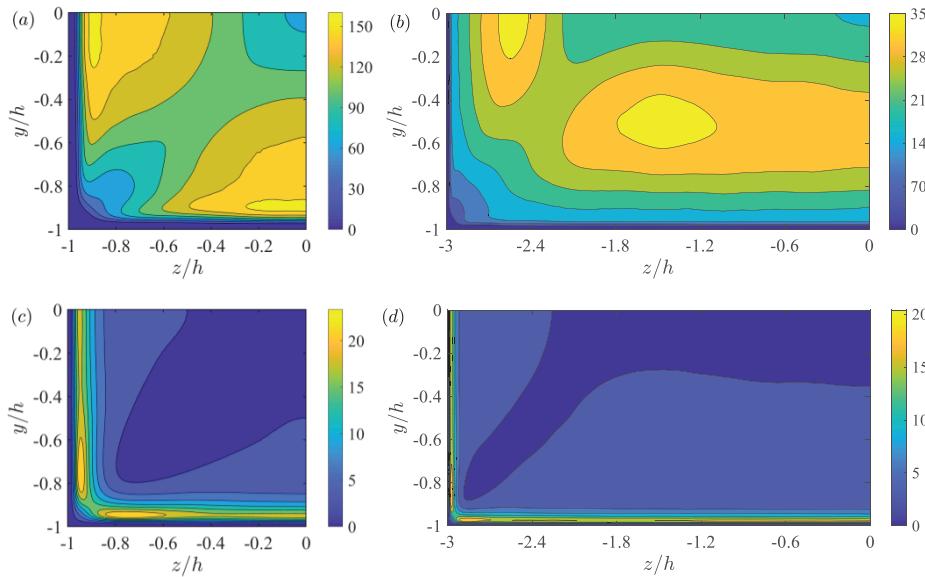


FIG. 4. Representative input features (a), (b) Re_t and (c), (d) s_m for duct flows at (a), (c) $Re_\tau = 180$ with $AR = 1$ and (b), (d) $Re_\tau = 360$ with $AR = 3$. Only the lower left quadrant of the duct is shown due to the symmetry of the configuration. z and y denote the directions parallel to the horizontal and vertical walls.

(e.g., secondary flow) and extensive engineering interest. Six DNS datasets at various Re_τ and aspect ratios (ARs) are used:^{103–105} (i) $AR = 1, 3, 5, 7$ at $Re_\tau = 180$ and (ii) $AR = 1, 3$ at $Re_\tau = 360$. Both the training and test datasets are from these two flows.

To utilize data from different sources, the raw data should be pre-processed using a unified standard. Thus, we reformat the data and compute the input feature Re_t . For channel flows, all raw data are nondimensionalized by u_τ and the viscous length scale ν/u_τ (signified with +); thus, $k^+ = k/u_\tau^2$, $\varepsilon^+ = \nu\varepsilon/u_\tau^4$, and $Re_t \equiv k^2/(\nu\varepsilon) = k^{+2}/\varepsilon^+$ (see Fig. 3). For duct flows, all raw data are normalized by the bulk velocity u_b and h (signified with *); thus, $k^* = k/u_b^2$ and $\varepsilon^* = h\varepsilon/u_b^3$. Accordingly, $Re_t = Re_b k^{*2}/\varepsilon^*$, where $Re_b \equiv u_b h/\nu$ is the bulk Reynolds number. Other input features and the targeted stress anisotropy are computed similarly. Representative input features Re_t and s_m for duct flows are shown in Fig. 4. Note that $s_m = \omega_m$ in channel flows and $s_m \approx \omega_m$ in duct flows, so the redundant feature ω_m can be omitted.

B. Clustering-based unbiased sampling technique

It is common knowledge that ML-based training can exhibit a preference toward patterns that are shared by more training data, especially when adopting the commonly used mean squared error as the cost function. A “pattern” that covers more data accumulates

larger errors and thus has a bigger impact; in return, this pattern is assigned a high priority by the error backpropagation, e.g., TBNN-1 in Fig. 2(b). Thus, an unbiased strategy is needed to ensure equal sample diversity in the training data.

A k-means clustering algorithm¹⁰⁶ is employed as an unsupervised classifier to assign the data into groups without *prior* knowledge. Considering that the stress anisotropy is targeted in the cost function, the stress anisotropy is selected as a variable for clustering. Consequently, the k-means algorithm automatically categorizes the whole domain into two different subregions, as shown in Fig. 5, resulting in similar patterns within the same cluster, i.e., nearly homogeneous flows far from the wall and nonhomogeneous flows in the near-wall region. This clustering result, unsurprisingly, reflects the effects of wall proximity and can be explained using layered eigenvalues ($\lambda_1, \lambda_2, \lambda_3$) of the stress anisotropy. The spatial distribution of each eigenvalue in Fig. 6 is consistent with the clustering result in Fig. 5. The major role of the k-means algorithm is to quantitatively characterize the boundary between two subregions, which cannot be qualitatively extracted from Fig. 6.

For two-dimensional channel flows, the clustering boundary is approximately located at the point of the maximum shear stress, with a scaling law of $Re_\tau^{3/10}$. Chen *et al.*¹⁰⁷ theoretically derived a scaling law of $Re_\tau^{1/3}$ for this location. The above discussion suggests that a

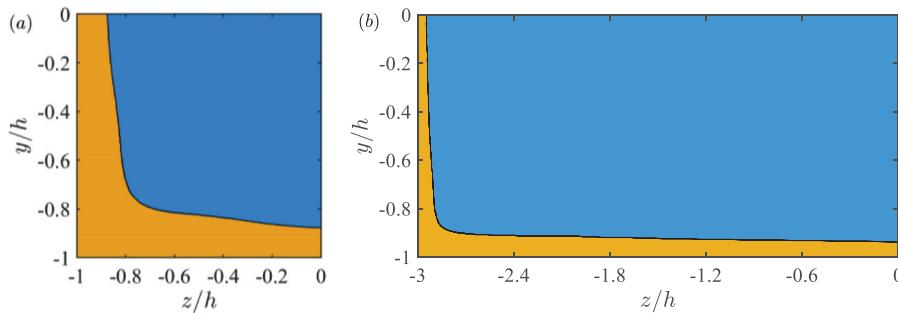


FIG. 5. Clustering results for duct flows at (a) $Re_\tau = 180$ with $AR = 1$ and (b) $Re_\tau = 360$ with $AR = 3$. The whole domain is separated into two subregions: regions far from the wall (blue) and near the wall (yellow).

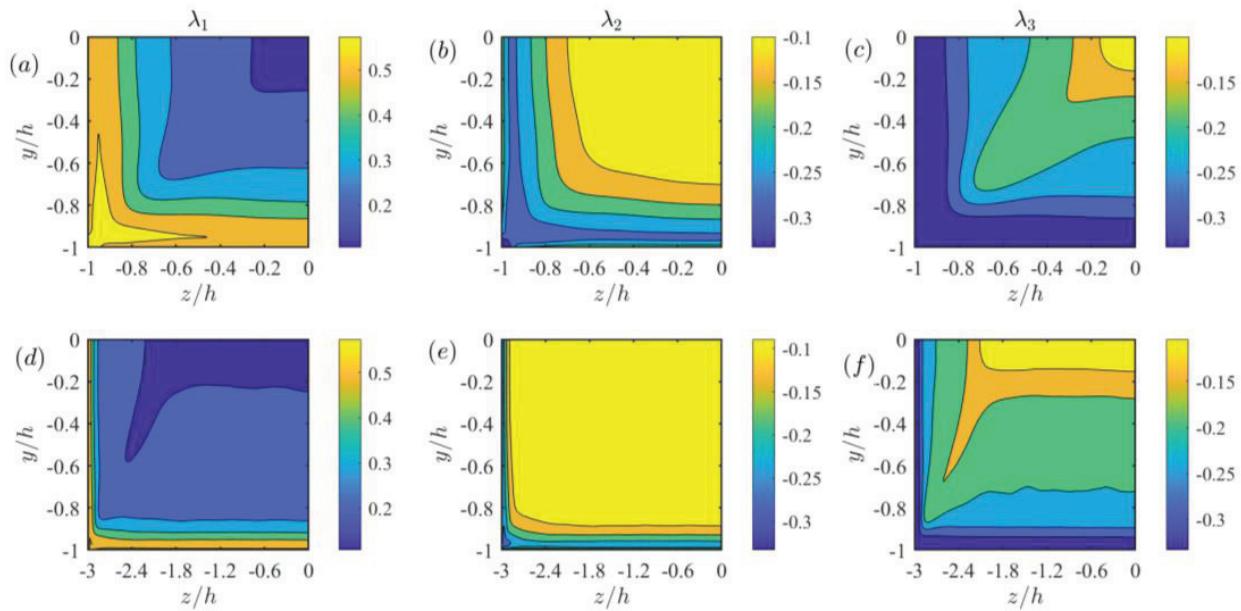


FIG. 6. Eigenvalues ($\lambda_1, \lambda_2, \lambda_3$) of the stress anisotropy for duct flows at (a)–(c) $Re_\tau = 180$ with $AR = 1$ and (d)–(f) $Re_\tau = 360$ with $AR = 3$. The left, middle, and right columns correspond to λ_1, λ_2 , and λ_3 , respectively. Only the lower left quadrant of the duct is shown due to the symmetry of the configuration.

clustering-based data sampling technique for data-driven turbulence modeling is not only a mathematical operation but also has a physical meaning. Accordingly, an oversampling technique based on this clustering approach is developed to rebalance the sample diversity between two clustered subregions, thus guaranteeing that different flow patterns are treated fairly in the training process. Specifically, a

refined interpolation is applied in the clustered subregion that has fewer data samples until the number of samples in this subregion is approximately equal to that of the other. Note that the clustering-unbiased sampling technique herein will be very useful in developing a universal data-driven model across multiple classes of flows in the near future.

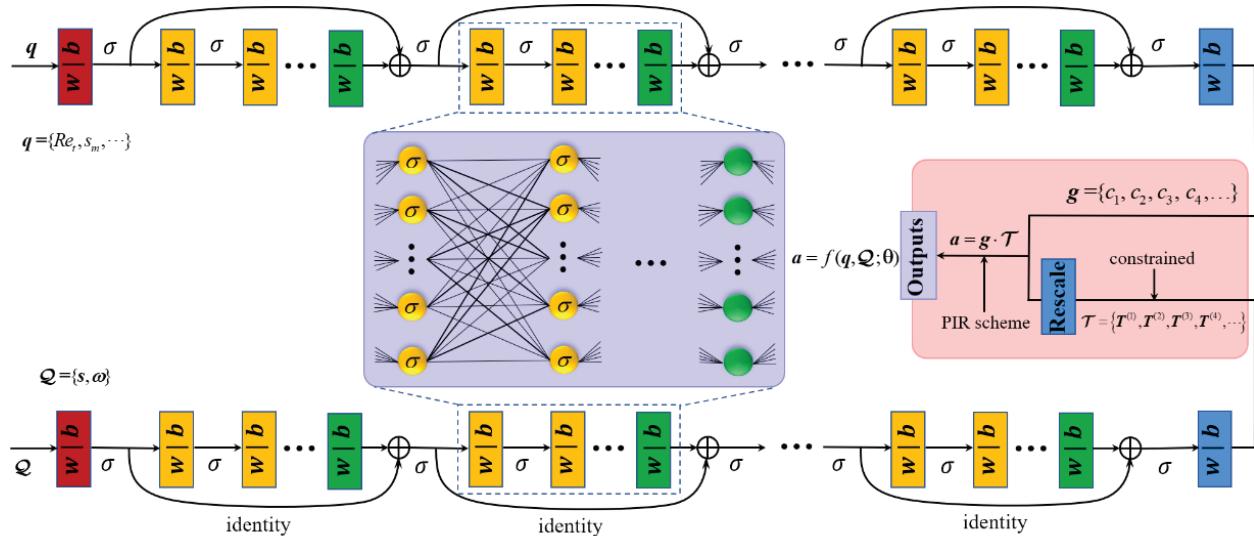


FIG. 7. Schematic of PiResNet based on a residual network with fully connected layers. All operations in the neural network are element-wise. A fair and robust training strategy for PiResNet is detailed in Algorithm 2.

C. Neural network and training experiments

Under the proposed UIML shown in Fig. 1, $\mathbf{q} = \{Re_t, s_m, \dots\}$ and $\mathcal{Q} = \{\mathbf{s}, \omega\}$ are used as input features. A specialized physics-informed residual network (PiResNet) for this UIML is designed to represent the overall mapping $f : \{\mathbf{q}, \mathcal{Q}\} \mapsto \mathbf{a}$ based on the training dataset \mathcal{D} , as depicted in Fig. 7. PiResNet is an acyclic cascade consisting of an input layer, several hidden layers, and an output layer. A skip connection is applied across several hidden layers to provide an identity mapping, thus forming a residual block with the operation $F(\mathbf{x}_i) + \mathbf{x}_i \mapsto \mathbf{x}_{i+m}$, where $F(\cdot)$ is the overall operation of m fully connected layers contained in each residual block. In a fully connected layer that has a nonlinear activation function σ , $\sigma(\mathbf{w}_i \mathbf{x}_i + \mathbf{b}_i) \mapsto \mathbf{x}_{i+1}$, where \mathbf{w}_i and \mathbf{b}_i are network parameters. He *et al.*¹⁰⁸ and Chang *et al.*¹⁰⁹ demonstrated the superiority of the identity mappings in overcoming two technical problems that are frequently encountered in deep learning: gradient vanishing and performance degradation. Considering that only plain neural networks have been used in turbulence closure modeling to date, the first use of residual neural networks enriches our understanding of DNN-based turbulence modeling. Accordingly, the modeling problem is cast as an optimization problem by minimizing the following cost function:

$$\mathcal{J}(\boldsymbol{\theta}; \lambda_w, \lambda_c; \mathcal{D}) \equiv \mathcal{L}_a + \lambda_w \mathcal{L}_w + \lambda_c \mathcal{L}_c, \quad (17a)$$

ALGORITHM 2: PiResNet Training Scheme. All operations are element-wise.

MAIN

Inputs: Training dataset \mathcal{D} , learning rate η , and weighting factors (λ_w, λ_c) .

Initialization: Set $t = 0$;

Randomly initialize $\boldsymbol{\theta}_t = [\mathbf{w}, \mathbf{b}]$.

Iteration: $\mathcal{J}(\boldsymbol{\theta}_t) \leftarrow \text{FORWARD}(\mathbf{q}, \mathcal{Q}; \boldsymbol{\theta}_t)$;

while not convergent **do**

 (*cross-validation as a stopping criteria*)

$t \leftarrow t + 1$;

$\boldsymbol{\theta}_t \leftarrow \text{BACKWARD}(\mathbf{q}, \mathcal{Q}; \boldsymbol{\theta}_{t-1})$;

$\mathcal{J}(\boldsymbol{\theta}_t) \leftarrow \text{FORWARD}(\mathbf{q}, \mathcal{Q}; \boldsymbol{\theta}_t)$;

end

Outputs: Return resulting $\boldsymbol{\theta}$ as the optimal $\boldsymbol{\theta}_\pi$.

$$\begin{aligned} \mathcal{L}_a \in & \left\{ \mathcal{L}_a^1 \equiv \frac{1}{9N} \sum \|\mathbf{a} - \tilde{\mathbf{a}}\|_F^2, \mathcal{L}_a^2 \equiv \frac{1}{9N} \sum \|\gamma \odot (\mathbf{a} - \tilde{\mathbf{a}})\|_F^2, \right. \\ & \left. \mathcal{L}_a^3 \equiv \frac{1}{9N} \sum \|\mathbf{a} \tilde{\mathbf{a}}^{-1} - \mathbf{I}\|_F^2 \right\}, \end{aligned} \quad (17b)$$

$$\mathcal{L}_w \equiv \|\mathbf{w}\|_2^2, \quad \mathcal{L}_c \equiv \frac{1}{4N} \sum \|\mathbf{c}\|_2^2, \quad (17c)$$

where $\boldsymbol{\theta} = [\mathbf{w}, \mathbf{b}]$ are trainable parameters, N is the number of samples in \mathcal{D} , λ_w and λ_c are regularization factors, and γ is a fair-factor matrix. \mathcal{L}_a represents supervised data-driven aspects, for which we provide three alternatives in Eq. (17b) to achieve an unbiased prediction for the stress anisotropy components. \mathcal{L}_w represents the complexity of the network; thus, $\lambda_w \neq 0$ to prevent overfitting and obtain noise-insensitive closure coefficients. The regularization constraint on \mathcal{L}_c is intended to ease ill-posed problems and reduce the noise-sensitivity of the learned model to the structural bases. Both $\lambda_w \neq 0$ and $\lambda_c \neq 0$, as well as the PIR scheme constrained on \mathbf{a} , are unsupervised physics-informed aspects embedded in the training phase. Finally, the Adam algorithm¹¹⁰ is adopted to compute the cost function gradients, and the backpropagation algorithm¹¹¹ is used to update the network parameters. The training procedure is summarized in Algorithm 2, and serves as a fair and robust training strategy to achieve optimal parameters $\boldsymbol{\theta}_\pi$; thus, $\mathbf{a} = f(\mathbf{q}, \mathcal{Q}; \boldsymbol{\theta}_\pi)$.

FORWARD ($\mathbf{q}, \mathcal{Q}; \boldsymbol{\theta}$)

Require: $\mathbf{q}, \mathcal{Q}, \tilde{\mathbf{a}}, \lambda_w, \lambda_c; \boldsymbol{\theta}$.

$\mathbf{a} \leftarrow f(\mathbf{q}, \mathcal{Q}; \boldsymbol{\theta})$ as Fig. 7;

 Compute $\mathcal{J}(\boldsymbol{\theta})$ by Eq. (17)

end

BACKWARD ($\mathbf{q}, \mathcal{Q}; \boldsymbol{\theta}$)

Require: $\mathbf{q}, \mathcal{Q}, \tilde{\mathbf{a}}, \lambda_w, \lambda_c, \eta; \boldsymbol{\theta}$.

 Compute $\nabla \mathcal{J}(\boldsymbol{\theta})$ using Adam;

 (*gradient with bias-corrected momentum*)

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla \mathcal{J}(\boldsymbol{\theta})$. // update rule

end

The numeric range of input features and outputs should be limited in order to ease the updating of trainable network parameters. A logarithmic function $\log(1 + x)$ is selected to rescale all input features, as this makes Re_t work better. In comparison, the learned model responses are more sensitive to noise in the input features when adopting commonly used normalization with global means and variances, which should therefore be avoided. Note that the benefits of logarithmic functions may go far beyond feature scaling. Weatheritt and Sandberg²¹ found that the logarithm has successfully evolved in each best regression solution when using an evolutionary algorithm. Previous work by Jiang *et al.*⁹ demonstrated that closure coefficients have a separation of order of magnitude. Thus, the structural bases are

also rescaled as shown in Fig. 7. Another concern is the selection of activation functions, which is based on the following considerations: (i) smooth functions (infinitely differentiable) to account for the smoothness requirement of the predicted stress anisotropy and (ii) sensitivity to negative values to prevent dead neurons. Thus, the Gaussian error linear unit (GELU)¹¹² is chosen as the activator; in our testing, this outperforms both the rectified linear unit (ReLU)¹¹³ and exponential linear unit (ELU).¹¹⁴ GELU achieves smaller network weights, thus less sensitive to noise in the input features. Additionally, the comparative testing of a plain neural network (16 hidden layers) and a residual neural network (10 hidden layers) indicates that the latter achieves better and more robust predictions. Thus, the residual

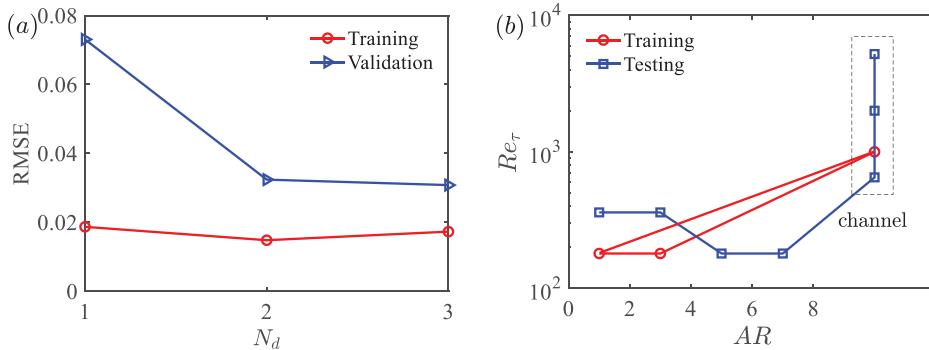


FIG. 8. (a) Training and validation errors (root mean squared error, RMSE) as functions of the training dataset size (N_d) and (b) flow configurations adopted for PiResNet training and testing.

neural network is adopted as the DNN architecture. The resulting network has an optimal structure with five residual blocks, each containing two fully connected hidden layers with 128 neurons in each layer.

The effect of the training dataset size on the performance of PiResNet is investigated through cross-validation, as shown in Fig. 8(a). Accordingly, three flow cases are used for training: a channel flow at $Re_\tau = 1000$ and two duct flows with $AR = 1$ at $Re_\tau = 180$ and $AR = 3$ at $Re_\tau = 360$; the remaining cases are used for testing, as shown in Fig. 8(b). Note that trained PiResNet using only the duct flow with $AR = 1$ at $Re_\tau = 180$ still achieves a good extrapolation to the duct flow with $AR = 3$ at $Re_\tau = 360$, but slightly underpredicts the shear stresses for a duct flow with $AR = 7$ at $Re_\tau = 180$. The additional data better capture the effect of the large aspect ratio.

To compare the fair prediction ability in two clustering subregions, both originally imbalanced data and clustering-based rebalanced data are used to train PiResNet (denoted as PiResNet-1 and PiResNet-2, respectively). The discrepancy in the prediction errors between the two subregions is presented in Table I, where a smaller value indicates a fairer prediction. The ratio of the sample numbers in the two subregions, r_s , is defined to characterize the degree of data imbalance. PiResNet-1, trained with the imbalance data, tends to produce an unfair prediction, and this unfairness increases with increasing r_s . In comparison, PiResNet-2 provides fairer predictions for different flow domains: a more than 40% improvement in the training dataset and an almost 25% improvement in the testing dataset, with a 69% rise for the case with $AR = 1$ at $Re_\tau = 360$. Thus, the clustering-based sampling technique is effective.

Similarly, we further examine whether PiResNet achieves fair predictions of the stress anisotropy components. Equation (17b) provides three different cost functions for the first term of Eq. (17a), i.e., \mathcal{L}_a^1 , \mathcal{L}_a^2 , and \mathcal{L}_a^3 . In particular, the fair factor in \mathcal{L}_a^2 is properly selected by the trial-and-error method as $\gamma_{ij} = \sqrt{N / \sum |\bar{a}_{ij}|}$. The comparative

result is presented in Table II. \mathcal{L}_a^1 gives biased predictions for a_{11} and a_{12} , whereas \mathcal{L}_a^2 and \mathcal{L}_a^3 produce fairer predictions of all stress anisotropy components. In particular, the prediction of a_{23} is significantly improved. However, \mathcal{L}_a^2 is not frame-invariant and depends on the matrix γ (not a tensor). For the same flow as in the training case, PiResNet with \mathcal{L}_a^2 provides different predictive accuracy when different meshes are used. Thus, we choose \mathcal{L}_a^3 to form a scale- and frame-invariant cost function. Consequently, the hyperparameters remain valid across training datasets of dynamically similar flows that have different scales or different coordinate systems.

Furthermore, we search optimal values of λ_w and λ_c from 10^{-3} – 10^{-7} , eventually selecting $\lambda_w = 5 \times 10^{-7}$ and $\lambda_c = 4 \times 10^{-6}$. In this setting, PiResNet has smaller values of both the overall network parameters and closure coefficients, as shown in Fig. 9. When the regularization constraint is only applied to the closure coefficients ($\lambda_w = 0$, $\lambda_c \neq 0$), the network parameters are hardly limited [see Fig. 9(b)]; whereas when the regularization constraint is only applied to network parameters ($\lambda_w \neq 0$, $\lambda_c = 0$), there is a limited impact on closure coefficients [see Fig. 9(c)]. Thus, it is necessary to apply regularization constraints to both the network parameters and closure coefficients. It is worth noting that the regularization of the closure coefficients is critical from two aspects. First, the ill-posed problem of the cost function gradients being insensitive to the closure coefficients may occur when the strain is zero somewhere in the flow. Thus, we introduce additional information to regularize the solution by penalizing the total magnitude of the closure coefficients. To some extent, this also increases the smoothness. Second, the closure coefficients can be regarded as amplifiers of the uncertainty of the structural bases. Smaller closure coefficients reduce the noise-sensitivity of the learned model to the structural bases. Similarly, smaller network parameters mean lower noise-sensitivity of strain-dependent closure coefficients. Section IV C further illustrates their quantitative effects.

TABLE I. Discrepancy of root mean squared errors between two clustering subregions. The comparison of both models was conducted using the same training flows at almost the same level of the overall training error.

	AR = 1 $Re_\tau = 180$	AR = 3 $Re_\tau = 360$	AR = 1 $Re_\tau = 360$	AR = 3 $Re_\tau = 180$	AR = 5 $Re_\tau = 180$	AR = 7 $Re_\tau = 180$
r_s	3.07	3.17	2.68	3.28	3.43	3.61
PiResNet-1	5.97×10^{-3}	8.85×10^{-3}	3.33×10^{-3}	1.96×10^{-2}	2.01×10^{-2}	2.05×10^{-2}
PiResNet-2	3.22×10^{-3}	5.17×10^{-3}	1.02×10^{-3}	1.45×10^{-2}	1.53×10^{-2}	1.57×10^{-2}

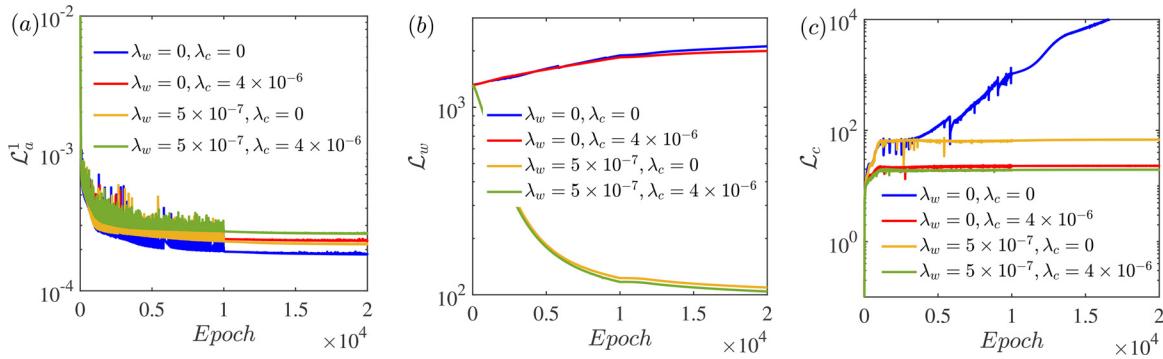


FIG. 9. Learning dynamics of PiResNet with four sets of hyperparameters (λ_w, λ_c): (a) \mathcal{L}_a^1 , representing the quality of data-driven training; (b) \mathcal{L}_w , representing the complexity of the network; and (c) \mathcal{L}_c , representing the magnitude of the closure coefficients. All training experiments were performed with \mathcal{L}_a^3 and evaluated with \mathcal{L}_a^1 . \mathcal{L}_c was computed using rescaled closure coefficients due to the rescaling of the structural bases.

IV. TESTING: NUMERICAL RESULTS AND VERIFICATION

In this section, as outlined in Fig. 1, the PiResNet-predicted outcomes are systematically evaluated from different aspects (e.g., accuracy, realizability, and robustness) to drive the trust toward PiResNet. For two classes of test flows, the roles that the Reynolds stress plays in the mean-flow field are different: the shear stress dominates the flow in a channel,¹⁰⁷ whereas the normal stress controls the secondary flow in a duct.¹¹⁵ It is recognized that conventional closure models lack accuracy, both for near-wall flow in a channel and cross-plane flow in a duct. Thus, the *prior* and *posterior* performances of PiResNet are validated by examining its ability to capture the aforementioned flow characteristics and their sensitivities to the Reynolds number and aspect ratio.

To illustrate the superiority of PiResNet, we also compare it to the same-level model based on gene expression programming²¹ (denoted as GEP). As stressed by Musgrave *et al.*,¹¹⁶ it is not fair to compare ML models to conventional models. Besides, DNN (as the kernel of PiResNet) and GEP are different representatives of modern ML algorithms: the former uses parametric regression, while the latter uses symbolic regression. These are why we take GEP as the baseline. This is the first attempt, to our knowledge, to compare different ML-based turbulence models.

A. Predictions of stress anisotropy

The *prior* process is centered around whether the well-trained PiResNet can successfully provide satisfactory predictions of unseen

TABLE II. Correlation coefficients for three different \mathcal{L}_a defined in Eq. (17b) corresponding to \mathcal{L}_a^1 , \mathcal{L}_a^2 , and \mathcal{L}_a^3 , respectively. Results for all training and testing flows are given.

	a	a_{11}	a_{22}	a_{33}	a_{12}	a_{13}	a_{23}	
Training	\mathcal{L}_a^1	0.955	0.972	0.959	0.967	0.945	0.924	0.689
	\mathcal{L}_a^2	0.958	0.969	0.968	0.971	0.946	0.942	0.883
	\mathcal{L}_a^3	0.956	0.969	0.968	0.969	0.944	0.945	0.886
Testing	\mathcal{L}_a^1	0.928	0.951	0.925	0.931	0.922	0.908	0.590
	\mathcal{L}_a^2	0.933	0.939	0.938	0.940	0.923	0.920	0.789
	\mathcal{L}_a^3	0.932	0.938	0.937	0.938	0.921	0.925	0.794

flows, i.e., its extrapolation capability. The PiResNet-predicted stress anisotropy for channel flows at $Re_\tau = 650, 2000$, and 5200 is shown in Fig. 10. The PiResNet results are consistent with DNS in all stress anisotropy components. In comparison, GEP only provides satisfactory predictions for the shear stress, and only in the region far from the wall, which is a slight advantage over conventional nonlinear models in this flow. Moreover, GEP, unsurprisingly, returns the same stress for the near-wall region as for the region far from the wall due to the inappropriate turbulent timescale. Introducing Re_t , which parameterizes the newly proposed turbulent timescale, enables PiResNet to extend the resolution scope to the whole flow domain. Thus, the concept of Eq. (12) is effective.

The PiResNet-predicted stress anisotropy for duct flows at $Re_\tau = 360$ with $AR = 1$ is shown in Fig. 11, and that at $Re_\tau = 180$ with $AR = 7$ is shown in Fig. 12. For brevity, the results for duct flows with smaller aspect ratios (i.e., $AR = 3$ and 5) at $Re_\tau = 180$ are omitted. Due to the symmetry of the configuration, only the lower left quadrant of the duct ($-AR \leq z/h \leq 0, -1 \leq y/h \leq 0$) is shown. PiResNet clearly provides improved predictions of all stress components, while GEP only gives relatively accurate predictions for shear stresses a_{12} and a_{13} . The unfair prediction of the normal stresses by GEP comes from its biased training cost function. It is common knowledge that the cross-flow normal stress imbalance is of critical importance in capturing secondary motions in duct flows. Additionally, GEP lacks accuracy in the near-wall region. In comparison, PiResNet correctly captures the secondary mechanism and displays a good predictive ability for near-wall flows. PiResNet can be successfully transferred to flows with different Reynolds numbers and aspect ratios, indicating that PiResNet learns the underlying similar dynamical mechanism. It is worth noting that a fair training strategy enables PiResNet to improve the predictions of a_{23} .

B. Realizability capability

A barycentric map (BMap)⁸² is used to examine whether the PIR corrector really functions toward achieving a realizable solution. As an undistorted representation of anisotropy, BMap linearly relates any anisotropy state to three limiting states, i.e.,

$$\boldsymbol{\pi} = (\lambda_1 - \lambda_2)\boldsymbol{\pi}_{1c} + 2(\lambda_2 - \lambda_3)\boldsymbol{\pi}_{2c} + (3\lambda_3 + 1)\boldsymbol{\pi}_{3c}, \quad (18)$$

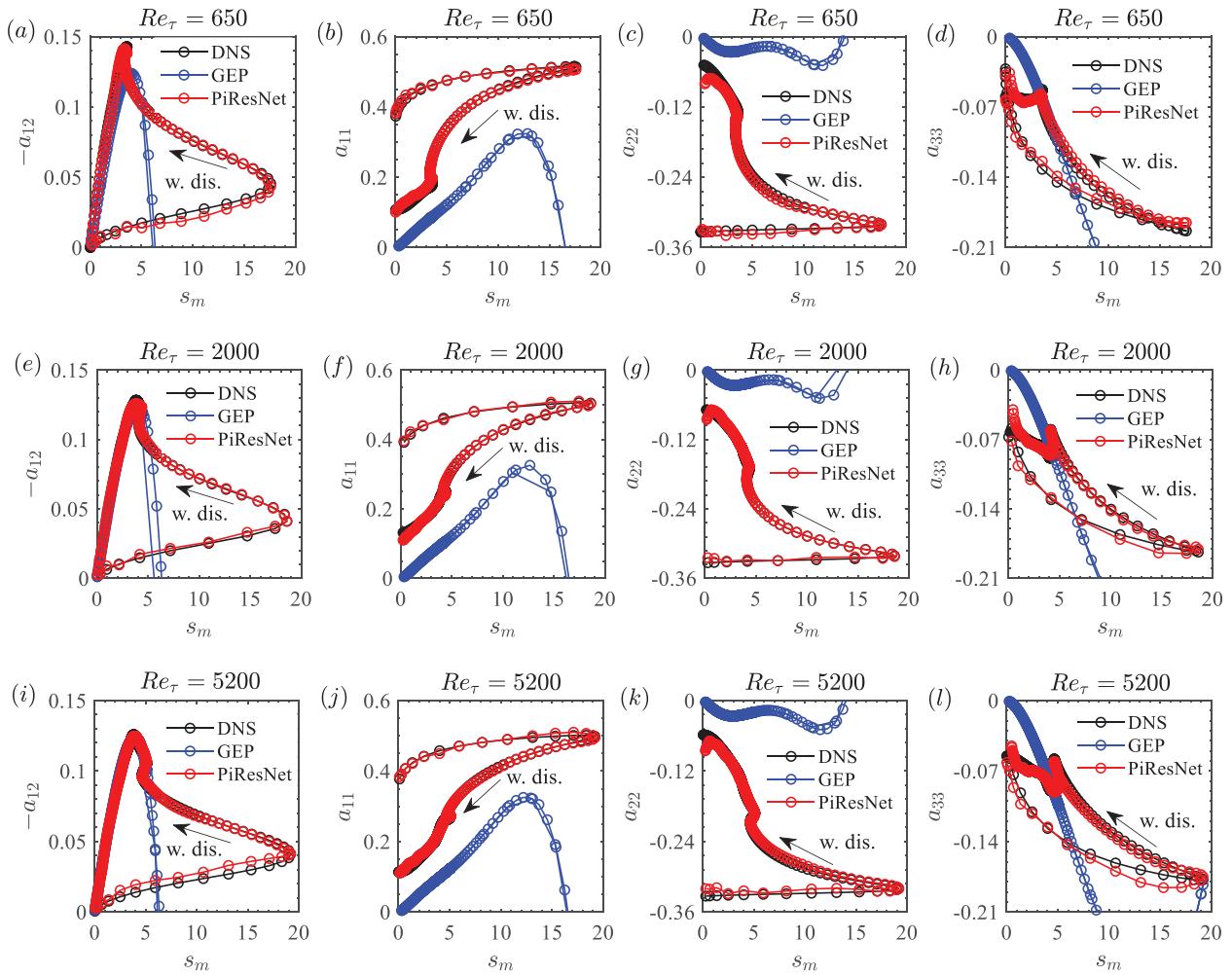


FIG. 10. Variation of stress anisotropy components (a_{ij}) with nondimensional shear parameter (s_m) predicted by GEP, PiResNet, and DNS for channel flows at (a)–(d) $Re_\tau = 650$, (e)–(h) $Re_\tau = 2000$, and (i)–(l) $Re_\tau = 5200$. Columns 1–4 correspond to a_{12} , a_{11} , a_{22} , and a_{33} , respectively. The direction of increasing wall distance (w. dis.) is also marked. Note that some GEP results are out of range and not shown.

where π_{1c} , π_{2c} , and π_{3c} denote the three vertices of BMap in representing the one component state (1C), two-component axisymmetric state (2C-axis), and three-component isotropic state (3C-iso).

Comparative results for channel flows at $Re_\tau = 650$, 2000, and 5200 are shown in Fig. 13. PiResNet achieves good agreement with DNS and is successfully constrained within the well-defined BMap; however, GEP is mismatched with DNS, especially in the region far from the wall. GEP shows a more obvious bias toward the 3C-iso state than PiResNet, thus indicating a poor characterization of anisotropy.

Further results for duct flows at $Re_\tau = 360$ with $AR = 1$ and $Re_\tau = 180$ with $AR = 7$ are shown in Figs. 14 and 15. PiResNet consistently provides realizable solutions that agree well with DNS. In comparison, realizable anisotropy is only obtained by GEP in the region far from the wall. In particular, GEP returns to a 3C-iso state closer to the wall. As demonstrated by Weatheritt and Sandberg,²¹ this qualitative incorrectness is due to inappropriate use of the timescale. With Eq. (12), our new timescale τ_I never tends to zero as the wall is

approached. When abandoning the PIR corrector [see Figs. 14(a) and 15(a)], our proposed PiResNet provides realizable solutions in most flow regions other than the near-wall region. To further demonstrate the effectiveness of the PIR corrector and its advantage over that of Ling *et al.*,²⁴ both realizability correctors are applied to the GEP-predicted stress anisotropy [see Figs. 14(d)–14(f) and 15(d)–15(f)]. Consequently, our proposed PIR corrector easily helps GEP regain realizable solutions, while that of Ling *et al.*²⁴ still fails. Thus, the PIR corrector is effective and can be extended to other turbulence models to ensure realizability.

C. Robustness performance

The robustness of PiResNet is investigated in terms of its noise-sensitivity. Additional Gaussian noise is added to the input features in the form

$$\varphi' = \varphi(1 + \delta\ell), \quad \ell \sim \mathcal{N}(0, 1), \quad (19)$$

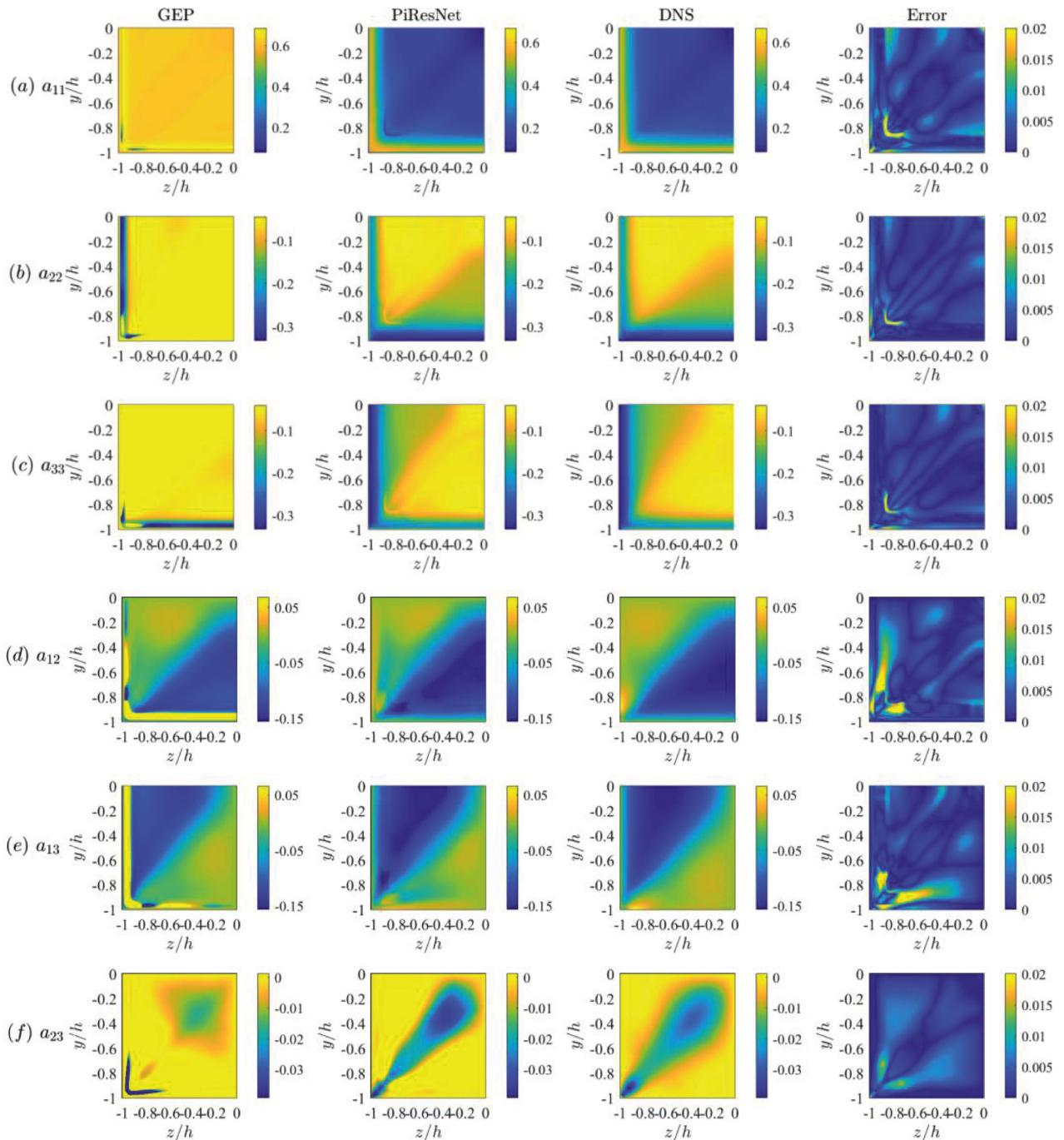


FIG. 11. Stress anisotropy components on the duct cross-plane predicted by GEP, PiResNet, and DNS for a duct flow at $Re_\tau = 360$ with AR = 1: (a) a_{11} , (b) a_{22} , (c) a_{33} , (d) a_{12} , (e) a_{13} , and (f) a_{23} . Columns 1–4 correspond to the GEP, PiResNet, and DNS results and the absolute error between PiResNet and DNS, respectively. Note that this case is an extrapolation both of the Reynolds number and aspect ratio. For simplicity, only the lower left quadrant of the duct ($-1 \leq z/h, y/h \leq 0$) is shown. The coordinate origin is located at the duct center.

where φ' and φ denote the input features with and without noise, δ is the noise level, and ℓ is a set of random numbers that satisfies a Gaussian distribution. The relative error is defined as

$$\epsilon_a = \langle \|\mathbf{a} - \tilde{\mathbf{a}}\|_F / \|\tilde{\mathbf{a}}\|_F \rangle, \quad (20)$$

where $\langle \cdot \rangle$ denotes averaging over the spatial flow domain. Figure 16 shows the variation in the relative prediction errors of PiResNet with

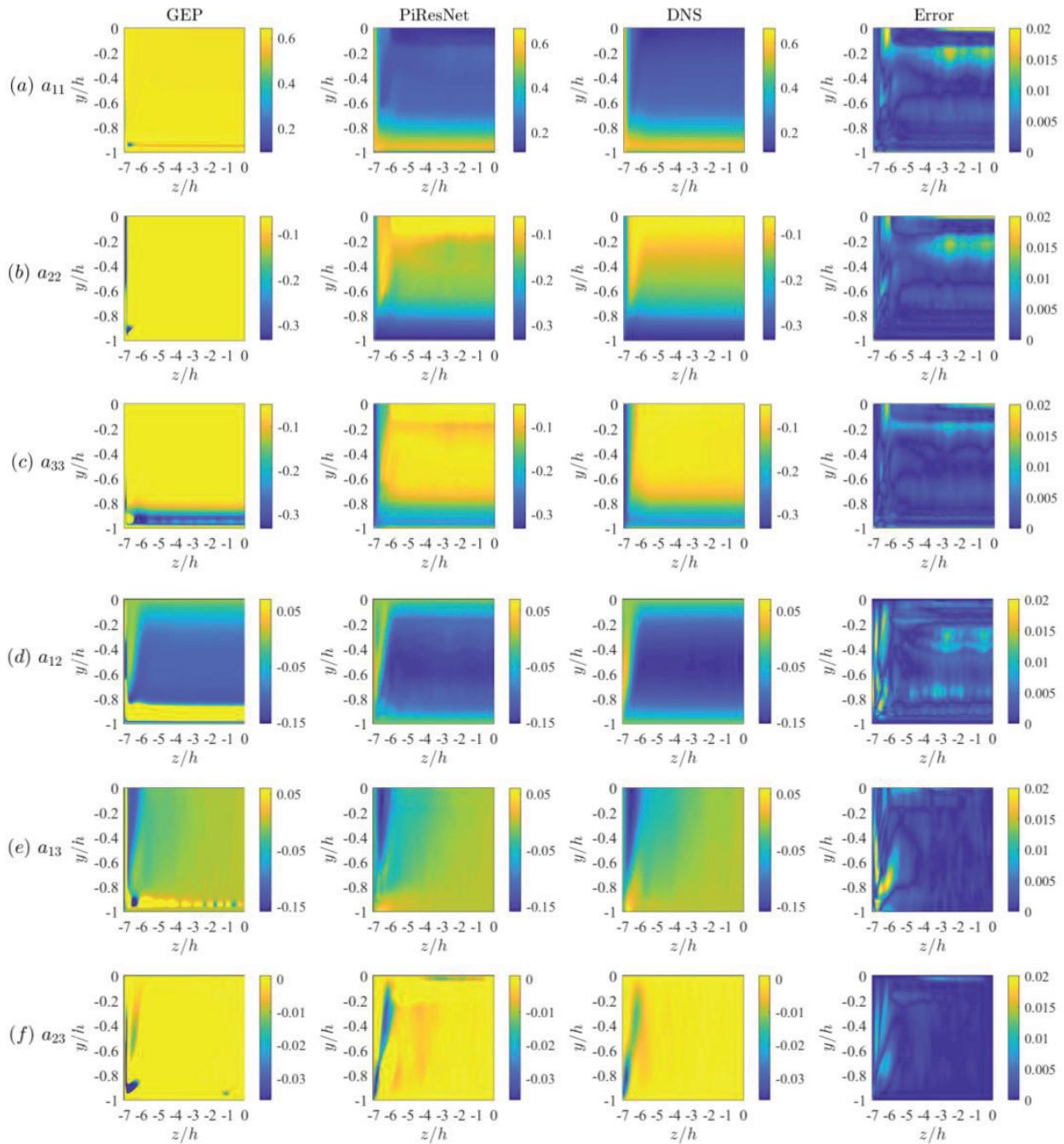


FIG. 12. Stress anisotropy components on the duct cross-plane predicted by GEP, PiResNet, and DNS for duct flow at $Re_t = 180$ with $AR = 7$: (a) a_{11} , (b) a_{22} , (c) a_{33} , (d) a_{12} , (e) a_{13} , and (f) a_{23} . Columns 1–4 correspond to the GEP, PiResNet, and DNS results and the absolute error between PiResNet and DNS, respectively. Note that this case is a large extrapolation both of the aspect ratio. For simplicity, only the lower left quadrant of the duct ($-7 \leq z/h, y/h \leq 0$) is shown. The coordinate origin is located at the duct center.

various noise levels added to the input features. Three activators with the same regularization operation are compared. The prediction error is almost unchanged up to a noise level of 30%, and PiResNet with GELU is the most insensitive to perturbations in the input features.

Thus, PiResNet displays good noise immunity, which is of critical importance to numerical simulations. This is also very useful in improving the experimental Reynolds stress when combining PiResNet with noisy experimental data.

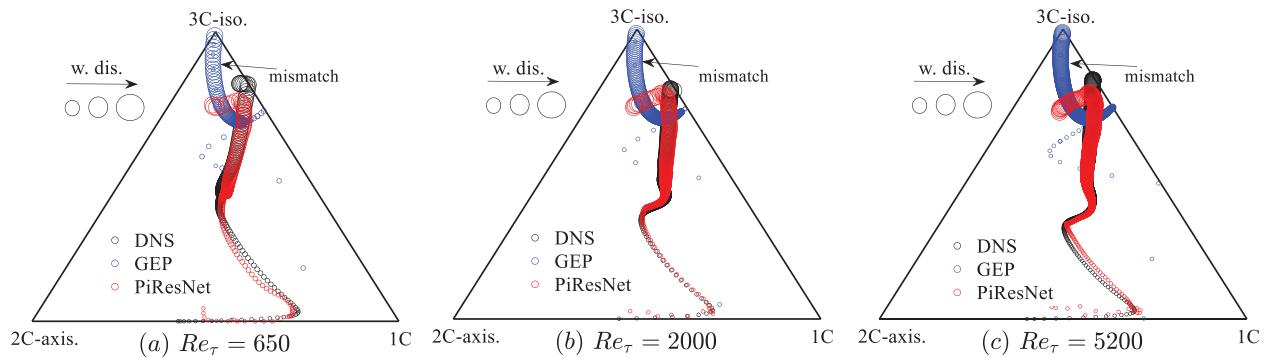


FIG. 13. Scatter plot of the stress anisotropy in the barycentric map for channels flow at (a) $Re_\tau = 650$, (b) $Re_\tau = 2000$, and (c) $Re_\tau = 5200$, scaled with the corresponding wall distance (w. dis.). Results of GEP and PiResNet are compared with DNS. Note that some GEP results are out of range and not shown.

As mentioned previously, the robustness of PiResNet stems from two factors: (i) the noise-sensitivity of the strain-dependent closure coefficients depends on the network parameters, and (ii) the noise-sensitivity of the learned model to the structural bases depends on the closure coefficients. The neuron-averaged value of 0.025 is obtained for the network parameters, thus resulting in noise-insensitive closure coefficients. The closure coefficients are presented in Table III and Fig. 17 (the closure coefficients of GEP are somewhat larger than those of PiResNet, and thus are omitted). Although a complete set of structural bases has been considered, only the first four terms of our PiResNet are required to reproduce the flow characteristics well in all test cases. Weatheritt and Sandberg²⁰ reported that more terms produce limited

gains in predictive accuracy. Thus, only four coefficients are shown. PiResNet provides a mean value of -0.077 for c_1 , which is close to the -0.09 of conventional models. As is the case for GEP, large values of the closure coefficients appear in PiResNet on the duct bisector due to the vanishing velocity gradients. The difference is that the penalty on the PiResNet-predicted closure coefficients [as shown in Eq. (17a)] helps to regularize the solution of ill-posed problems, and so much smaller values than GEP can be successfully obtained. The coefficients (c_1, c_2, c_3, c_4) of PiResNet have small values (less than 0.18), which contributes to the noise-sensitivity of PiResNet to structural bases. Additionally, the small dispersion of the closure coefficients (see Table III) indicates good spatial smoothness. Larger values and a larger

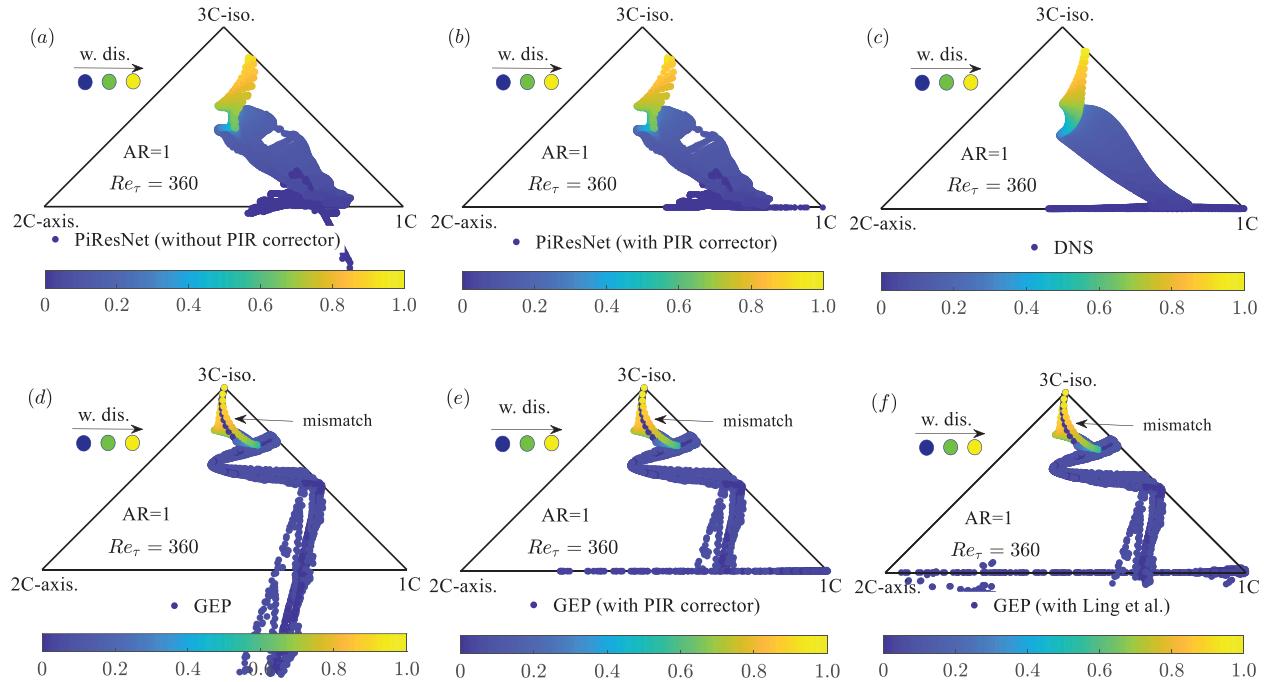


FIG. 14. Scatter plot of the stress anisotropy in the barycentric map for a duct flow at $Re_\tau = 360$ with $AR = 1$, colored by the corresponding wall distance (w. dis.): (a)–(b) PiResNet, (c) DNS, and (d)–(f) GEP.

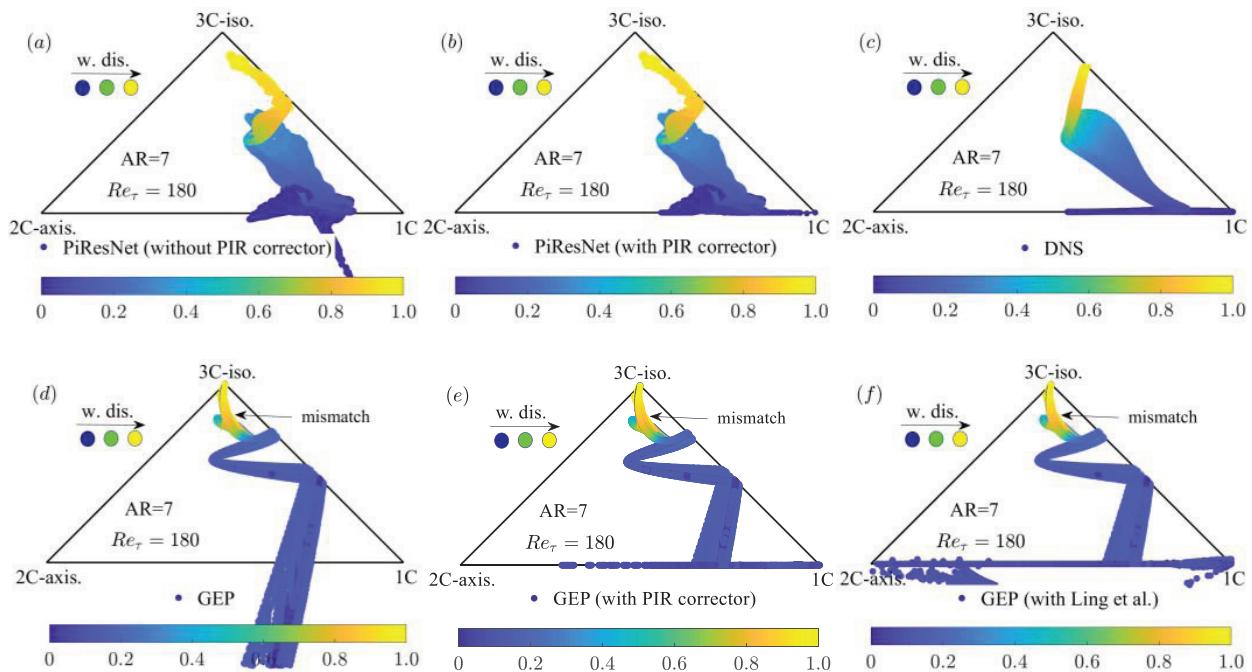


FIG. 15. Scatter plot of the stress anisotropy in the barycentric map for a duct flow at $Re_\tau = 180$ with $AR = 7$, colored by the corresponding wall distance (w. dis.): (a)–(b) PiResNet, (c) DNS, and (d)–(f) GEP.

dispersion of the closure coefficients occur when removing the regularizations of the closure coefficients. Obviously, regularization operations applied to both the network parameters and closure coefficients work well in improving the robustness of PiResNet.

D. Roles of closure coefficients

The closure coefficients are interpreted in terms of their physical roles, numerical aspects, and their contribution to the production of turbulent kinetic energy $\mathcal{P}_k \equiv -\boldsymbol{\tau} : \mathbf{S}$. In PiResNet $c_1 < 0$, while $c_1 > 0$ in GEP as the wall is approached. It is common experience that it is useful to have $c_1 < 0$ in numerical stability, which has been theoretically confirmed by a recent finding of Wu *et al.*¹¹⁷ A value of $c_2 > 0$ in

PiResNet indicates that s^2 plays the opposite role to the linear term s , while $\mathbf{s}\omega - \omega\mathbf{s}$ with $c_3 < 0$ plays the same role as s . In most regions, $c_4 < 0$, although $c_4 > 0$ in the narrow corner region. Accordingly, a negative a_{kl} is dominated by c_1 , while a positive a_{kl} is mainly controlled by c_2 . The well-known in-plane secondary motion is almost accounted for by c_2 , c_3 , and c_4 because $a_{22} - a_{33} \approx (c_2 + 2c_3 - c_4)(s_{12}^2 - s_{13}^2)/2$, which drives high-energy fluid from the near-wall region to the duct core. The dependency of $a_{22} - a_{33}$ on c_2 , c_3 , and c_4 is a *priori* idealization. In fact, this is not always the case. For instance, it is shown hereafter that $a_{22} - a_{33}$ is mainly contributed from c_3 in our well-trained PiResNet (see Sec. V A). Notably, the symmetric distribution of (c_1, c_2, c_3, c_4) about the duct bisector at $Re_\tau = 360$ with $AR = 1$, in agreement with the underlying flow mechanics associated with competitive effects of two adjacent walls, is a physical reflection of the embedded invariances in PiResNet. Furthermore, Eq. (14) in incompressible flows yields the PiResNet-predicted \mathcal{P}_k as

$$\mathcal{P}_k / \varepsilon = -2c_1\text{tr}(\mathbf{s}^2) - 2c_2\text{tr}(\mathbf{s}^3) - 2c_4\text{tr}(\mathbf{s}\omega - \omega\mathbf{s}). \quad (21)$$

The disappearance of c_3 from Eq. (21) but its existence in Eq. (14) implies that $\mathbf{s}\omega - \omega\mathbf{s}$ plays a redistribution role between stress

TABLE III. Spatial average (Ave.) and root mean square (RMS) of PiResNet-predicted closure coefficients for all test flows.

	c_1	c_2	c_3	c_4
Ave.	-0.077	0.036	-0.050	-0.024
RMS	0.041	0.035	0.041	0.042

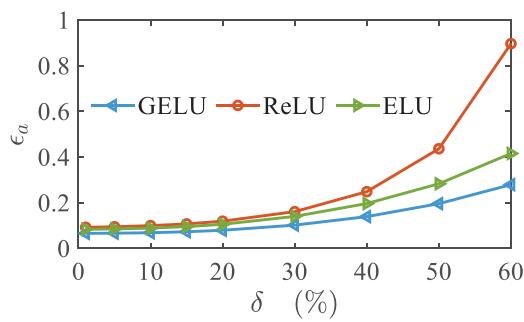


FIG. 16. Variation of PiResNet-predicted relative error (ϵ_a) on the test flows with increasing noise level (δ) in input features. Three activators are compared with the same regularization constraints on both the network parameters and closure coefficients.

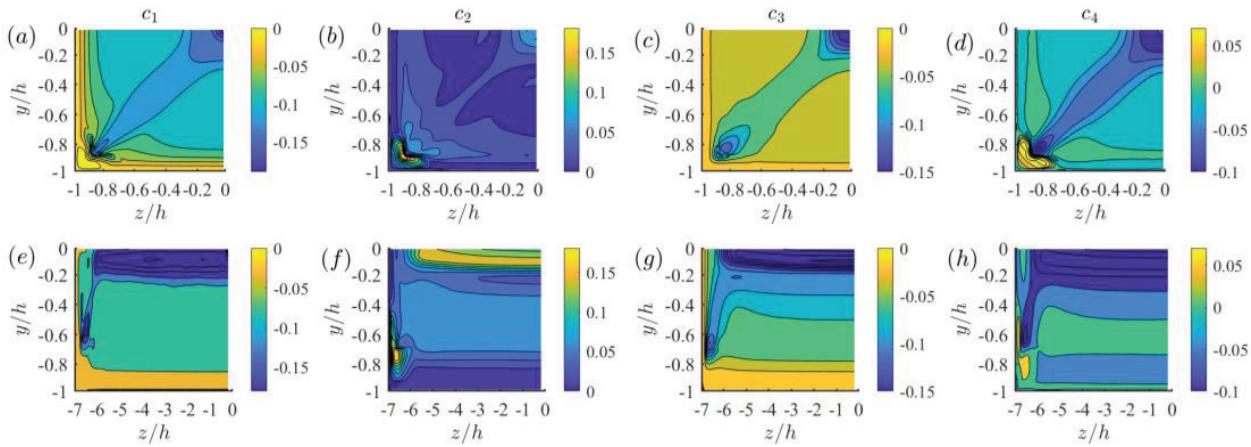


FIG. 17. PiResNet-predicted closure coefficients for a duct flow at (a)–(d) $Re_\tau = 360$ with $AR = 1$ and (e)–(h) $Re_\tau = 180$ with $AR = 7$. Columns 1–4 correspond to c_1 , c_2 , c_3 , and c_4 , respectively.

components and makes no contribution to \mathcal{P}_k , which is similar to the mechanism of pressure-strain correlations in transport equations. In duct flows, $\text{tr}(\mathbf{s}^2) > 0$, $\text{tr}(\mathbf{s}^3) < 0$, $\text{tr}(\boldsymbol{\omega}^2 \mathbf{s}) > 0$ (negative in the corner region) and $\text{tr}(\mathbf{s}^2) > \text{tr}(\mathbf{s}^3) > |\text{tr}(\boldsymbol{\omega}^2 \mathbf{s})|$. Thus, \mathbf{s} , \mathbf{s}^2 , and $\boldsymbol{\omega}^2$ in PiResNet accelerate the production of turbulent kinetic energy, while the linear term \mathbf{s} with $c_1 > 0$ in GEP, as the wall is approached, acts to hinder the production of turbulent kinetic energy. This behavior of GEP leads to a laminarization or “lift off” of the secondary structures.²¹ To achieve physical and converged results, Weatheritt and Sandberg²¹ claimed that an empirical modification of the GEP-predicted \mathcal{P}_k was vital. Our PiResNet provides a suitable mechanism for \mathcal{P}_k . Accordingly, the closure coefficients in PiResNet have clear groundings in physical arguments.

E. Prediction of mean-flow pattern

As analyzed before, PiResNet is a successful model with realizable and robust predictions that can account for the variability of flow characteristics due to changes in the Reynolds number and aspect ratio. To examine whether these encouraging gains would be passed on to numerical predictions of QoIs (e.g., the bulk and secondary flow

velocities), a *posteriori* validation is further performed in a CFD environment. Thus, PiResNet is inserted into the incompressible RANS equations, as shown in Fig. 1. Specifically, the linear part of PiResNet, $\mathbf{a}^L \equiv c_1 \mathbf{s}$, is preserved separately as an implicit term of the RANS equations, thus leaving $\mathbf{a}^\perp \equiv \mathbf{a} - \mathbf{a}^L$ as an explicit source. Different from commonly under-relaxing \mathbf{a} against the isotropic Boussinesq’s formula with an *ad hoc* blending parameter, this implicit treatment with inherent $c_1 < 0$ has a more rigorous physical justification in improving the conditioning of discretized RANS equations, as recently detailed by Wu *et al.*¹¹⁷ The production term of the k -equation is approximated by Eq. (21), rather than the Boussinesq’s formula. The open-source CFD toolbox simpleFOAM¹¹⁸ is adopted as a steady-state flow solver with structured meshes. Nonslip boundary conditions are used at the walls and periodic conditions are applied in the streamwise direction. The wall resolution is such that $y^+ < 1$ for the first off-wall cell.

Figure 18 shows the predicted axial mean velocity profiles U/U_c for channel flows, where U_c is the mean velocity at the channel centerline. The PiResNet predictions almost overlap with the DNS results, while there is a small discrepancy between GEP and DNS. This is due to the overpredicted U^+ by GEP in the outer layer and its decreasing

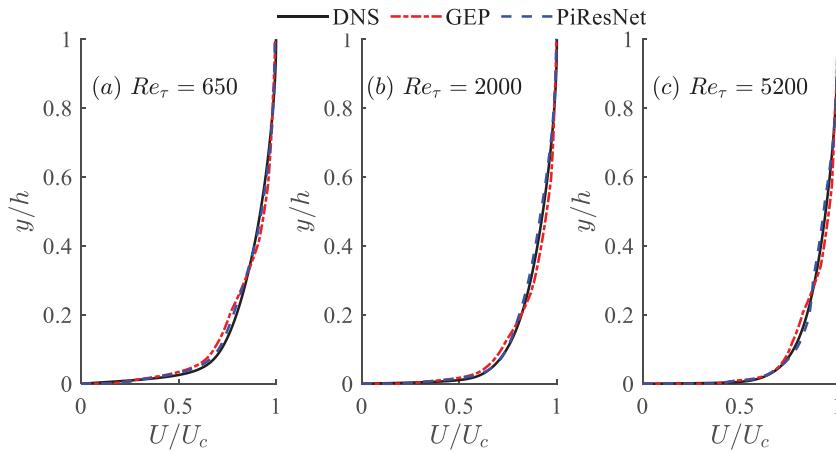


FIG. 18. Axial mean velocity U/U_c for channel flows at (a) $Re_\tau = 650$, (b) $Re_\tau = 2000$, and (c) $Re_\tau = 5200$. Both GEP and PiResNet predictions are compared with reference DNS. U_c is the channel centerline velocity.

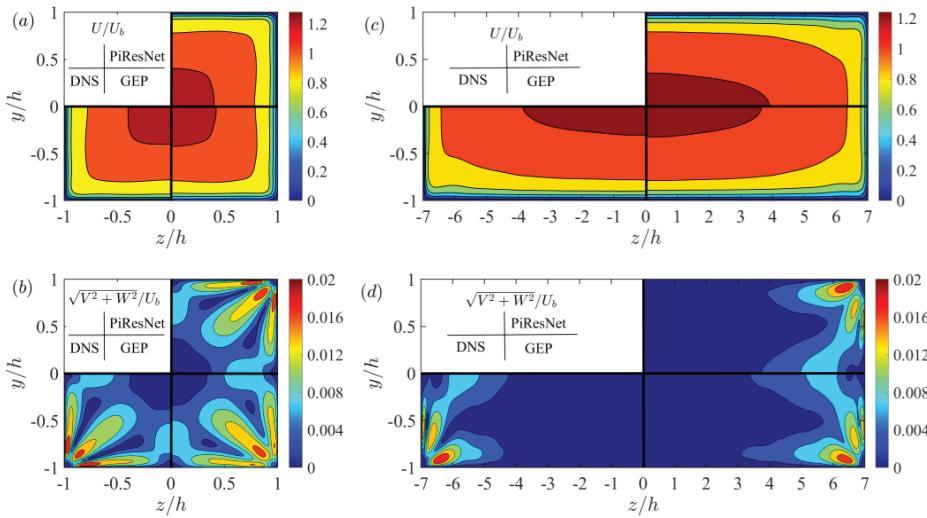


FIG. 19. Predicted mean streamwise velocity U/U_b and secondary velocity magnitude $\sqrt{V^2 + W^2}/U_b$ of two extrapolative duct flows: (a), (b) $Re_\tau = 360$ with $AR = 1$; (c), (d) $Re_\tau = 180$ with $AR = 7$. GEP and PiResNet predictions are compared with DNS. U_b is the bulk velocity.

slope, indicating that GEP has no obvious superiority over conventional models in channel flows (see also Fig. 10).

Figure 19 further compares predictions in duct flows, and the velocity profiles are detailed in Fig. 20, where U , V , and W are, respectively, the mean velocities in the streamwise, vertical and spanwise directions (U_b is the duct bulk velocity). The cross-flow velocity magnitude, $\sqrt{V^2 + W^2}/U_b$, represents the strength of the secondary flow, which is valuable in engineering. The PiResNet-predicted mean-flow patterns are closer to those of DNS than GEP. Specifically, for a duct flow at $Re_\tau = 360$ with $AR = 1$, the contours of U/U_b given by GEP are not “square” enough due to the weak $\sqrt{V^2 + W^2}/U_b$. In comparison, three lobe peaks of the secondary flow magnitude are relatively well predicted by PiResNet.

For a duct flow at $Re_\tau = 180$ with $AR = 7$, PiResNet predicts a more reasonable extent for $\sqrt{V^2 + W^2}/U_b$. At two given locations in Fig. 20, the profiles of V and W offered by PiResNet are more consistent with those of DNS than GEP. Considering that GEP has been illustrated to outperform the conventional Speziale-Sarkar-Gatski (SSG) model⁷⁹ significantly in predictions of $\sqrt{V^2 + W^2}/U_b$ (see Ref. 18), our superior PiResNet thus has a greater advantage over conventional models. Note that GEP utilizes the empirically corrected P_k , rather than that originally predicted by GEP.²¹ Additionally, the RANS-related transport equations are corrected in GEP. Ling *et al.*²⁴

have demonstrated quantitatively that such a correction of the transport equations is necessary. More recently, Brener *et al.*¹¹⁹ highlighted that the Reynolds stress that is recomputed to satisfy the balance of the RANS equations, rather than that directly provided by the DNS dataset, can further reduce the error propagation to the mean-flow field. This finding, although having not been practiced in the data-driven turbulence modeling, presents a promising prospect. In light of these reasons, there is still room for improvement of PiResNet in future work.

V. DISCUSSION AND EXTENSION

A. Underlying rationale behind prediction

Data-driven learning is at the core of recent advances in turbulence modeling; however, it has not had a transformative impact. One of the main obstacles lies in how to achieve a trustworthy data-driven turbulence model: (i) whether a model produces accurate predictions and (ii) whether a model behaves in a reasonable way. The first question concerns the generalizable accuracy (i.e., extrapolation capability) of a data-driven model. As shown in Sec. IV, PiResNet-predicted outcomes are satisfactory in our test set. Most recent studies have centered around this point. In comparison, the need for interpretability arising from the second question has rarely been considered. However, as highlighted by Kutz,⁶⁶ it is significantly important to understand what guarantees good outcomes. Thus, we further explore the underlying

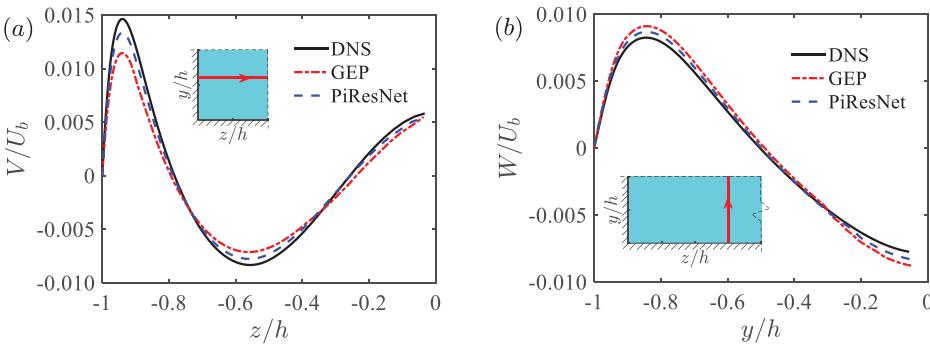


FIG. 20. Predicted secondary velocity profiles of two extrapolative duct flows: (a) V/U_b at $y/h = -0.4$ for $Re_\tau = 360$ with $AR = 1$; (b) W/U_b at $z/h = -5.5$ for $Re_\tau = 180$ with $AR = 7$. GEP and PiResNet predictions are compared with DNS. U_b is the bulk velocity.

rationale behind prediction and make it clear what PiResNet benefits from.

First, PiResNet benefits from domain knowledge embedded in the design and training phases. More specifically, Tables I and II indicate that unbiased principles improve the prediction favoritism toward a specific flow domain or stress component. The quantitative comparisons in Figs. 14 and 15 show that realizable states of PiResNet within BMap are guaranteed by the PIR corrector. The robustness performance of PiResNet, as shown in Fig. 16, originates from regularization constraints in Eq. (17a). The invariance requirements of PiResNet-predicted \mathbf{a} are satisfied by selecting invariants as inputs. Different from a *posterior* explanation, the effects of the embedded domain knowledge are definite and predictable before achieving a final predictive model.

Second, our resulting model behaves as a definite structure $\mathbf{a} = \sum g_i T^{(i)}$, thus indicating a linear combination of a set of structural bases. Since the Reynolds stress describes the rate and correlation of momentum transport in different directions of turbulent flows,⁶⁹ the structural bases can also be regarded as energy-related modes. The contribution to \mathbf{a} from each mode $T^{(i)}$ depends on g_i that reflects the effects of the boundary conditions. Thus, such a linear decomposition allows to interpret the underlying mechanism behind \mathbf{a} in a physical way. In addition, modeling turbulence physics by the structural and parametric parts is in accordance with common practices in the community of turbulence modeling. Accordingly, our framework has clear physical justifications.

Third, the sparsity constraint on $\mathbf{a} = \sum g_i T^{(i)}$ allows for a simple model to understand. The model accuracy improvement almost stagnates when there are more than four structural bases. Weatheritt and Sandberg²⁰ reported that more terms produce limited gains. In the initial training, the effective sparsity (i.e., a large sparsity penalty) leads to

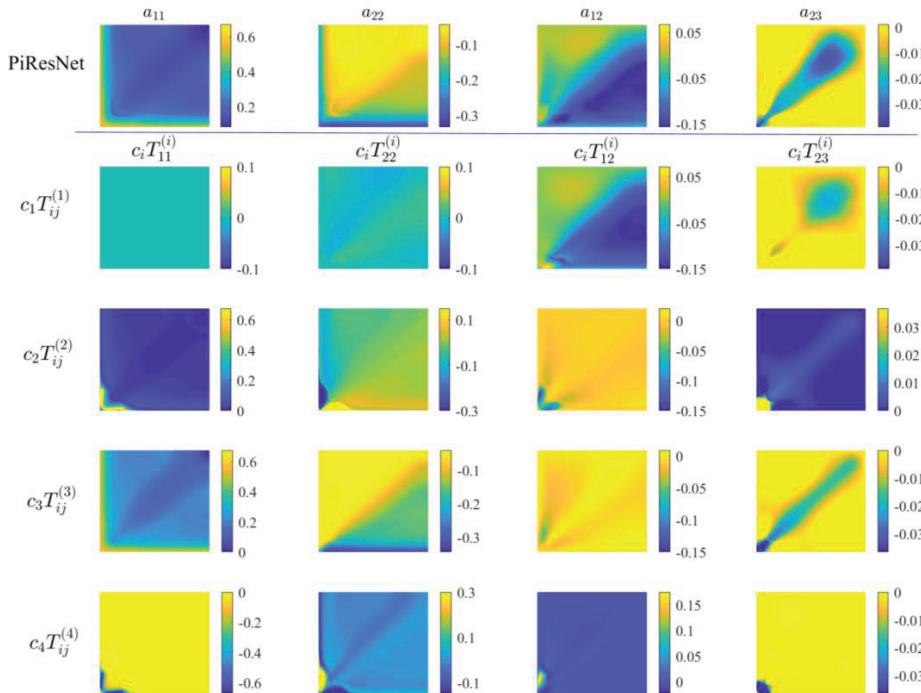


FIG. 21. Contributions to the stress anisotropy components a_{ik} from each constituent part $c_i T^{(i)}$ at $Re_\tau = 360$ with $AR = 1$, where c_i and $T^{(i)}$ correspond to the learned closure coefficients and structural bases. a_{33} and a_{13} are omitted due to a_{22} (a_{12}) and a_{33} (a_{13}) being symmetric about the duct bisector.

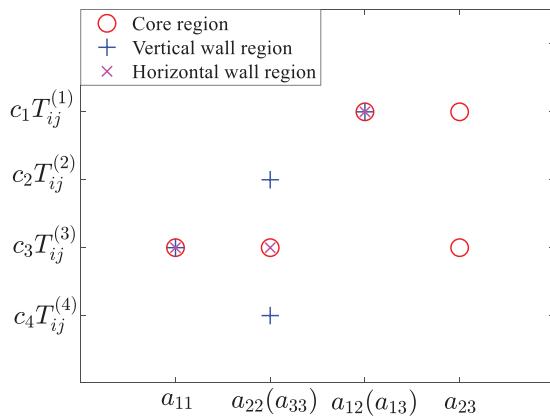


FIG. 22. Summary of the dominant contributions to the stress components a_{kl} from each constituent part $c_i T^{(i)}$ in different flow regions.

$(T_{33}^{(3)})$ has to be nonzero for a_{22} (a_{33}); while $T_{12}^{(1)}$ ($T_{13}^{(1)}$) has to be nonzero for a_{12} (a_{13}). To summarize, sparsity helps to identify a compact set of structural bases that are most important for capturing flow physics.

B. Potentials and limitations

A prior and *posterior* assessments have demonstrated the feasibility and merits of PiResNet. The underlying rationale behind prediction has been explored. However, due to the neglect of nonlocal effects, the predictive capability of PiResNet may degrade when extended to other untested complex flows with strong spatial inhomogeneity (e.g., rotation, curvature, massive separation). PiResNet and other recent landmark works^{21,24,27} are local models. The previous studies^{120,121} have shown that turbulence physics cannot be fully captured by such local models. Besides nonlocality, the memory effects are critical in non-equilibrium turbulence.^{122,123} There is a practical need to include such nonlocal and memory effects in data-driven turbulence modeling to achieve a universal and accurate model in the future.

It should be noted that the above-mentioned limitations are not inherent to our framework. Instead, these limitations are rooted in the feature selection of PiResNet, more specifically, an exclusion of pressure from input features. In turbulent flows, pressure is the dominant source of nonlocality in the Reynolds stress. Inspired by this, the inclusion of pressure-based variables in our framework helps to obtain a nonlocal model. The interested reader is referred to a recent work of Wu *et al.*⁴⁷ We also noticed that the direct-interaction approximation¹²⁴ and Biot–Savart integral¹²⁵ were successfully applied to conventional nonlocal modeling. Consequently, high-order velocity gradients arise in these nonlocal models. More recently, Jiang *et al.*⁵ also demonstrated the rationality of high-order velocity gradients. Thus, selecting these high-order terms as additional input features is a viable alternative. In principle, the above two nonlocal modeling approaches fall into a point-to-point mapping based on nonlocal variables as inputs. Besides, nonlocal modeling can be cast as a region-to-point mapping based on a convolutional neural network (CNN), inspired by the very early work of Chou¹²⁶ on Green's functions for pressure integral. Similarly, a network of Long Short-Term Memory (LSTM) can be used to account for memory effects of turbulence. One

may also be inspired by the early works of Speziale¹²⁷ and Taulbee¹²⁸ and thus include the Oldroyd (or material) derivative of the mean strain-rate tensor as input features. More recently, the effective strain theory of Hamlington and Dahn¹²⁹ demonstrated that such a dependence on the straining history can account for memory effects due to the Lagrangian variations in the mean strain rate. In summary, the above-mentioned nonlocal and nonequilibrium modeling approaches involve selecting input features and network architectures (e.g., CNN, LSTM), both of which can be easily integrated into our flexible framework. In turn, the merits of our framework can further enhance the vitality of these nonlocal and nonequilibrium modeling approaches.

It bears repeating that the goal of this work is not to propose a specific data-driven model, but to establish fundamental guidelines for the model development under our framework that go through the whole “design-training-testing-explanation” lifecycle. Closure modeling for nonlocal and memory effects is reserved for future work.

VI. SUMMARY AND PERSPECTIVES

The current work has developed a universal turbulence modeling framework under which an invariant, realizable, unbiased, and robust data-driven turbulence model was achieved. During the model development lifecycle, the underlying physical considerations are indispensable: domain knowledge has been attentively inferred and reasonably converted to modeling knowledge, embedded both in the design and training processes. In such endeavors, it is ensured that the proposed framework for turbulence modeling provides interpretable outcomes.

In the design phase, the objectivity of the proposed PiResNet was strictly preserved by choosing suitably extended Galilean-invariant input features. This principle prevents a preference for the coordinate system adopted by the modeler, which is user-friendly. Second, a dimensionless variable (Re_t) was introduced to parameterize the newly proposed turbulent timescale. As an additional argument, this successfully helped avoid possible nonunique mappings of conventional inputs to outputs of a RANS model. Third, as one of the contents of turbulence physics, realizability of PiResNet was respected using the PIR corrector, especially when the model is operating in an extrapolatory mode. It is worth noting that Re_t and the PIR corrector can also be applied to other algebraic RANS models.

In the training phase, a fair learning strategy comprising unbiased data sampling and unbiased cost function design was successfully implemented to update the network parameters. The unbiased data sampling based on a clustering technique is to ensure the equal diversity of training samples. The unbiased cost function design is twofold: (i) a scale- and frame-invariant cost function to ensure that the hyperparameter settings of PiResNet remain valid across dynamically similar training cases; and (ii) an effective measure to guarantee fairer contributions of the stress anisotropy components to the overall training error. Additionally, regularizations were applied to both the trainable network parameters and regressed closure coefficients to reduce the sensitivity of PiResNet to uncertainty in the input data, thus achieving a robust model.

It cannot be overemphasized that physical considerations are critical to data-driven turbulence modeling. The above-mentioned measures together improve the predictive performance of data-driven turbulence models based on limited data and are a significant step toward its practical use in the engineering environment in the near future. In addition, combining the noise-insensitive PiResNet and

noisy experimental data allows for developing a high-fidelity hybrid experiment-CFD method. While encouraging results have been achieved, there is much to be gained by further calibrating the RANS-related transport equations, enriching the diversity of training data, carefully exploring a broader set of input features, and including non-local and memory effects.

We do not claim that PiResNet is necessarily applicable to other untested complex flows although it does exhibit good generalization across some two- and three-dimensional flow configurations. PiResNet needs to be widely validated. Indeed, the philosophy and formalisms employed in the UIML are of a general nature, and are not restricted to a particular type of RANS-based closure, neural network architecture, or training case. Following the guidelines of UIML, one can easily extend this approach to retrain a data-driven model, even an LES formulation, to predict more complex flow configurations.

However, the most challenging issue is to develop a generalizable model across multiple classes of complex flows. Therefore, a benchmark dataset containing diverse typical flow phenomena (e.g., secondary effects, flow separation, streamline curvature) is urgently required for the development and evaluation of data-driven turbulence models. Additionally, a complete and compact set of input features should be systematically investigated. The incompleteness and redundancy may destroy the generalization capability. Finally, it is indeed essential to develop a “form-free-in-prior” data-driven model to directly infer the model form and model parameters so as to adequately represent the rich dynamics of turbulence, and that is the case with the UIML. This requires ML algorithms to deal directly with tensor problems, or at least vector problems (when using a spectral decomposition). Future work in this direction will be of great interest.

ACKNOWLEDGMENTS

This research was funded by the National Natural Science Foundation of China (Grant No. U1711265) and the Guangdong Science and Technology Department (Grant No. 2020B1212030001). We gratefully acknowledge Professor A. J. Giacomini of Queen’s University of Canada for the constructive comments on the first draft of the manuscript.

DATA AVAILABILITY

The data that support the findings of this study are available from the authors upon reasonable request.

REFERENCES

- ¹H. Choi and P. Moin, “Grid-point requirements for large eddy simulation: Chapman’s estimates revisited,” *Phys. Fluids* **24**, 011702 (2012).
- ²X. I. A. Yang and K. P. Griffin, “Grid-point and time-step requirements for direct numerical simulation and large-eddy simulation,” *Phys. Fluids* **33**, 015108 (2021).
- ³J. Jiménez, “Computing high-Reynolds-number turbulence: Will simulations ever replace experiments?,” *J. Turbul.* **4**, N22 (2003).
- ⁴J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis, “CFD vision 2030 study: A path to revolutionary computational aerosciences,” Report No. NASA/CR-2014-218178, 2014.
- ⁵K. Duraisamy, P. R. Spalart, and C. L. Rumsey, “Status, emerging ideas and future directions of turbulence modeling research in aeronautics,” Report No. NASA/TM-2017-219682, 2017.
- ⁶P. A. Durbin, “Some recent developments in turbulence closure modeling,” *Annu. Rev. Fluid Mech.* **50**, 77–103 (2018).
- ⁷P. R. Spalart, “Philosophies and fallacies in turbulence modeling,” *Prog. Aerosp. Sci.* **74**, 1–15 (2015).
- ⁸H. Xiao and P. Cinnella, “Quantification of model uncertainty in RANS simulations: A review,” *Prog. Aerosp. Sci.* **108**, 1–31 (2019).
- ⁹C. Jiang, J. Mi, S. Laima, and H. Li, “A novel algebraic stress model with machine-learning-assisted parameterization,” *Energies* **13**, 258 (2020).
- ¹⁰M. P. Brenner, J. D. Eldredge, and J. B. Freund, “Perspective on machine learning for advancing fluid mechanics,” *Phys. Rev. Fluids* **4**, 100501 (2019).
- ¹¹C.-W. Chang and N. T. Dinh, “Classification of machine learning frameworks for data-driven thermal fluid models,” *Int. J. Therm. Sci.* **135**, 559–579 (2019).
- ¹²K. Duraisamy, G. Iaccarino, and H. Xiao, “Turbulence modeling in the age of data,” *Annu. Rev. Fluid Mech.* **51**, 357–377 (2019).
- ¹³S. L. Brunton, B. R. Noack, and P. Koumoutsakos, “Machine learning for fluid mechanics,” *Annu. Rev. Fluid Mech.* **52**, 477–508 (2020).
- ¹⁴S. Pandey, J. Schumacher, and K. R. Sreenivasan, “A perspective on machine learning in turbulent flows,” *J. Turbul.* **21**, 567–584 (2020).
- ¹⁵B. Wang and J. Wang, “Application of artificial intelligence in computational fluid dynamics,” *Ind. Eng. Chem. Res.* **60**, 2772–2790 (2021).
- ¹⁶S. H. Cheung, T. A. Oliver, E. E. Prudencio, S. Prudhomme, and R. D. Moser, “Bayesian uncertainty analysis with applications to turbulence modeling,” *Reliab. Eng. Syst. Safe* **96**, 1137–1149 (2011).
- ¹⁷W. N. Edeling, P. Cinnella, and R. P. Dwight, “Predictive RANS simulations via Bayesian model-scenario averaging,” *J. Comput. Phys.* **275**, 65–91 (2014).
- ¹⁸W. N. Edeling, P. Cinnella, R. P. Dwight, and H. Bijl, “Bayesian estimates of parameter variability in the $k-\epsilon$ turbulence model,” *J. Comput. Phys.* **258**, 73–94 (2014).
- ¹⁹J. Zhang and S. Fu, “An efficient Bayesian uncertainty quantification approach with application to $k-\omega-\gamma$ transition modeling,” *Comput. Fluids* **161**, 211–224 (2018).
- ²⁰J. Weatheritt and R. Sandberg, “A novel evolutionary algorithm applied to algebraic modifications of the RANS stress-strain relationship,” *J. Comput. Phys.* **325**, 22–37 (2016).
- ²¹J. Weatheritt and R. D. Sandberg, “The development of algebraic stress models using a novel evolutionary algorithm,” *Int. J. Heat Fluid Flow* **68**, 298–318 (2017).
- ²²Y. Zhao, H. D. Akolekar, J. Weatheritt, V. Michelassi, and R. D. Sandberg, “RANS turbulence model development using CFD-driven machine learning,” *J. Comput. Phys.* **411**, 109413 (2020).
- ²³L. Y. Zhu, W. W. Zhang, J. Q. Kou, and Y. L. Liu, “Machine learning methods for turbulence modeling in subsonic flows around airfoils,” *Phys. Fluids* **31**, 015105 (2019).
- ²⁴J. Ling, A. Kurzawski, and J. Templeton, “Reynolds averaged turbulence modelling using deep neural networks with embedded invariance,” *J. Fluid Mech.* **807**, 155–166 (2016).
- ²⁵F. X. Trias, A. Gorobets, M. H. Silvis, R. W. C. P. Verstappen, and A. Oliva, “A new subgrid characteristic length for turbulence simulations on anisotropic grids,” *Phys. Fluids* **29**, 115109 (2017).
- ²⁶C. Xie, J. Wang, H. Li, M. Wan, and S. Chen, “Artificial neural network mixed model for large eddy simulation of compressible isotropic turbulence,” *Phys. Fluids* **31**, 085112 (2019).
- ²⁷S. Beetham and J. Capecelatro, “Formulating turbulence closures using sparse regression with embedded form invariance,” *Phys. Rev. Fluids* **5**, 084611 (2020).
- ²⁸A. Pal, “Deep learning emulation of subgrid-scale processes in turbulent shear flows,” *Geophys. Res. Lett.* **47**, e2020GL087005, <https://doi.org/10.1029/2020GL087005> (2020).
- ²⁹S. G. Rosofsky and E. Huerta, “Artificial neural network subgrid models of 2D compressible magnetohydrodynamic turbulence,” *Phys. Rev. D* **101**, 084024 (2020).
- ³⁰C. Xie, Z. Yuan, and J. Wang, “Artificial neural network-based nonlinear algebraic models for large eddy simulation of turbulence,” *Phys. Fluids* **32**, 115101 (2020).
- ³¹E. Dow and Q. Wang, “Quantification of structural uncertainties in the $k-\omega$ turbulence model,” AIAA Paper No. 2011-1762, 2011.
- ³²E. J. Parish and K. Duraisamy, “A paradigm for data-driven predictive modeling using field inversion and machine learning,” *J. Comput. Phys.* **305**, 758–774 (2016).

- ³³A. P. Singh and K. Duraisamy, "Using field inversion to quantify functional errors in turbulence closures," *Phys. Fluids* **28**, 045110 (2016).
- ³⁴A. P. Singh, S. Medida, and K. Duraisamy, "Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils," *AIAA J.* **55**, 2215–2227 (2017).
- ³⁵C. X. He, Y. Z. Liu, and L. Gan, "A data assimilation model for turbulent flows using continuous adjoint formulation," *Phys. Fluids* **30**, 105108 (2018).
- ³⁶M. Yang and Z. Xiao, "Improving the $k-\omega-\gamma-A_r$ transition model by the field inversion and machine learning framework," *Phys. Fluids* **32**, 064101 (2020).
- ³⁷M. Emory, J. Larsson, and G. Iaccarino, "Modeling of structural uncertainties in Reynolds-averaged Navier-Stokes closures," *Phys. Fluids* **25**, 110822 (2013).
- ³⁸C. Gorlé and G. Iaccarino, "A framework for epistemic uncertainty quantification of turbulent scalar flux models for Reynolds-averaged Navier-Stokes simulations," *Phys. Fluids* **25**, 055105 (2013).
- ³⁹G. Iaccarino, A. A. Mishra, and S. Ghili, "Eigenspace perturbations for uncertainty estimation of single-point turbulence closures," *Phys. Rev. Fluids* **2**, 024605 (2017).
- ⁴⁰R. L. Thompson, A. A. Mishra, G. Iaccarino, W. N. Edeling, and L. Sampaio, "Eigenvector perturbation methodology for uncertainty quantification of turbulence models," *Phys. Rev. Fluids* **4**, 044603 (2019).
- ⁴¹C. Gorlé, C. García-Sánchez, and G. Iaccarino, "Quantifying inflow and RANS turbulence model form uncertainties for wind engineering flows," *J. Wind Eng. Ind. Aerodyn.* **144**, 202–212 (2015).
- ⁴²C. Gorlé, S. Zeoli, M. Emory, J. Larsson, and G. Iaccarino, "Epistemic uncertainty quantification for Reynolds-averaged Navier-Stokes modeling of separated flows over streamlined surfaces," *Phys. Fluids* **31**, 035101 (2019).
- ⁴³A. A. Mishra, J. Mukhopadhyay, J. Alonso, and G. Iaccarino, "Design exploration and optimization under uncertainty," *Phys. Fluids* **32**, 085106 (2020).
- ⁴⁴A. A. Mishra and G. Iaccarino, "Theoretical analysis of tensor perturbations for uncertainty quantification of Reynolds averaged and subgrid scale closures," *Phys. Fluids* **31**, 075101 (2019).
- ⁴⁵H. Xiao, J.-L. Wu, J.-X. Wang, R. Sun, and C. J. Roy, "Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier-Stokes simulations: A data-driven, physics-informed Bayesian approach," *J. Comput. Phys.* **324**, 115–136 (2016).
- ⁴⁶J. 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).
- ⁴⁷J. L. Wu, H. Xiao, and E. Paterson, "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework," *Phys. Rev. Fluids* **3**, 074602 (2018).
- ⁴⁸J. L. Wu, R. Sun, S. Laizet, and H. Xiao, "Representation of stress tensor perturbations with application in machine-learning-assisted turbulence modeling," *Comput. Method Appl. Mech. Eng.* **346**, 707–726 (2019).
- ⁴⁹Y. Yin, P. Yang, Y. Zhang, H. Chen, and S. Fu, "Feature selection and processing of turbulence modeling based on an artificial neural network," *Phys. Fluids* **32**, 105117 (2020).
- ⁵⁰R. Maulik, O. San, A. Rasheed, and P. Vedula, "Data-driven deconvolution for large eddy simulations of Kraichnan turbulence," *Phys. Fluids* **30**, 125109 (2018).
- ⁵¹Z. Wang, K. Luo, D. Li, J. Tan, and J. Fan, "Investigations of data-driven closure for subgrid-scale stress in large-eddy simulation," *Phys. Fluids* **30**, 125101 (2018).
- ⁵²R. Maulik, O. San, J. D. Jacob, and C. Crick, "Sub-grid scale model classification and blending through deep learning," *J. Fluid Mech.* **870**, 784–812 (2019).
- ⁵³C. Xie, K. Li, C. Ma, and J. Wang, "Modeling subgrid-scale force and divergence of heat flux of compressible isotropic turbulence by artificial neural network," *Phys. Rev. Fluids* **4**, 104605 (2019).
- ⁵⁴C. Xie, J. Wang, and E. Weinan, "Modeling subgrid-scale forces by spatial artificial neural networks in large eddy simulation of turbulence," *Phys. Rev. Fluids* **5**, 054606 (2020).
- ⁵⁵Z. Yuan, C. Xie, and J. Wang, "Deconvolutional artificial neural network models for large eddy simulation of turbulence," *Phys. Fluids* **32**, 115106 (2020).
- ⁵⁶S. Yao, B. Wang, A. Kronenburg, and O. T. Stein, "Modeling of sub-grid conditional mixing statistics in turbulent sprays using machine learning methods," *Phys. Fluids* **32**, 115124 (2020).
- ⁵⁷T. Readshaw, T. Ding, S. Rigopoulos, and W. P. Jones, "Modeling of turbulent flames with the large eddy simulation–probability density function (LES–PDF) approach, stochastic fields, and artificial neural networks," *Phys. Fluids* **33**, 035154 (2021).
- ⁵⁸W. N. Edeling, G. Iaccarino, and P. Cinnella, "A return to eddy viscosity model for epistemic UQ in RANS closures," in *Annual Research Briefs* (Center for Turbulence Research, Stanford University, 2016), pp. 273–287.
- ⁵⁹K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of coordinates and governing equations," *Proc. Natl. Acad. Sci. U. S. A.* **116**, 22445–22451 (2019).
- ⁶⁰S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Sci. Adv.* **3**, e1602614 (2017).
- ⁶¹S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Natl. Acad. Sci. U. S. A.* **113**, 3932–3937 (2016).
- ⁶²M. 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).
- ⁶³H. Vaddireddy, A. Rasheed, A. E. Staples, and O. San, "Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensor observation data," *Phys. Fluids* **32**, 015113 (2020).
- ⁶⁴L. Zanna and T. Bolton, "Data-driven equation discovery of ocean mesoscale closures," *Geophys. Res. Lett.* **47**, e2020GL088376, <https://doi.org/10.1029/2020GL088376> (2020).
- ⁶⁵J. Zhang and W. Ma, "Data-driven discovery of governing equations for fluid dynamics based on molecular simulation," *J. Fluid Mech.* **892**, A5 (2020).
- ⁶⁶J. N. Kutz, "Deep learning in fluid dynamics," *J. Fluid Mech.* **814**, 1–4 (2017).
- ⁶⁷J. Ling, R. Jones, and J. Templeton, "Machine learning strategies for systems with invariance properties," *J. Comput. Phys.* **318**, 22–35 (2016).
- ⁶⁸S. B. Pope, "A more general effective-viscosity hypothesis," *J. Fluid Mech.* **72**, 331–340 (1975).
- ⁶⁹J. L. Lumley, "Toward a turbulent constitutive relation," *J. Fluid Mech.* **41**, 413–434 (1970).
- ⁷⁰J. A. Anderson, *An Introduction to Neural Networks* (MIT Press, Massachusetts, 1995).
- ⁷¹K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Las Vegas, USA, 2016), pp. 770–778.
- ⁷²Q. Liu and D. Wang, "Stein variational gradient descent: A general purpose Bayesian inference algorithm," in *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, USA, 2016), pp. 2378–2386.
- ⁷³C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nat. Mach. Intell.* **1**, 206–215 (2019).
- ⁷⁴A. Spencer and R. Rivlin, "Isotropic integrity bases for vectors and second-order tensors," *Arch. Ration. Mech. Anal.* **9**, 45–63 (1962).
- ⁷⁵R. W. Johnson, *Handbook of Fluid Dynamics* (CRC Press, Florida, 2016).
- ⁷⁶M. Alber, A. Buganza Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold, and E. Kuhl, "Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences," *npj Digital Med.* **2**, 115 (2019).
- ⁷⁷S. Pawar, O. San, B. Aksoylu, A. Rasheed, and T. Kvamsdal, "Physics guided machine learning using simplified theories," *Phys. Fluids* **33**, 011701 (2021).
- ⁷⁸S. B. Pope, *Turbulent Flows* (Cambridge University Press, New York, 2000).
- ⁷⁹C. G. Speziale, S. Sarkar, and T. B. Gatski, "Modelling the pressure-strain correlation of turbulence: An invariant dynamical systems approach," *J. Fluid Mech.* **227**, 245–272 (1991).
- ⁸⁰Y. N. Huang and F. Durst, "Reynolds stress under a change of frame of reference," *Phys. Rev. E* **63**, 056305 (2001).
- ⁸¹J. L. Lumley, "Computational modeling of turbulent flows," *Adv. Appl. Mech.* **18**, 123–176 (1979).

- ⁸²S. Banerjee, R. Krahl, F. Durst, and C. Zenger, "Presentation of anisotropy properties of turbulence, invariants versus eigenvalue approaches," *J. Turbul.* **8**, N32 (2007).
- ⁸³J. Jiménez, "Machine-aided turbulence theory," *J. Fluid Mech.* **854**, R1 (2018).
- ⁸⁴W. Noll, "A mathematical theory of the mechanical behavior of continuous media," *Arch. Ration. Mech. Anal.* **2**, 197–226 (1958).
- ⁸⁵G. Haller, "An objective definition of a vortex," *J. Fluid Mech.* **525**, 1–26 (2005).
- ⁸⁶N. Geneva and N. Zabaras, "Quantifying model form uncertainty in Reynolds-averaged turbulence models with Bayesian deep neural networks," *J. Comput. Phys.* **383**, 125–147 (2019).
- ⁸⁷M. Lee and R. D. Moser, "Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$," *J. Fluid Mech.* **774**, 395–415 (2015).
- ⁸⁸K. Hanjalic and B. E. Launder, "Contribution towards a Reynolds-stress closure for low-Reynolds-number turbulence," *J. Fluid Mech.* **74**, 593–610 (1976).
- ⁸⁹Z. Yang and T.-H. Shih, "New time scale based k- ϵ model for near-wall turbulence," *AIAA J.* **31**, 1191–1198 (1993).
- ⁹⁰T. Craft, B. Launder, and K. Suga, "Development and application of a cubic eddy-viscosity model of turbulence," *Int. J. Heat Fluid Flow* **17**, 108–115 (1996).
- ⁹¹U. Goldberg and D. Apsley, "A wall-distance-free low Re k- ϵ turbulence model," *Comput. Method Appl. Mech. Eng.* **145**, 227–238 (1997).
- ⁹²M. M. Rahman, R. K. Agarwal, M. J. Lampinen, and T. Siikonen, "Improvements to the Rahman-Agarwal-Siikonen one-equation turbulence model based on k- ϵ closure," AIAA Paper No. 2014-3331, 2014.
- ⁹³P. A. Durbin, "Near-wall turbulence closure modeling without 'damping functions,'" *Theor. Comp. Fluid Dyn.* **3**, 1–13 (1991).
- ⁹⁴P. A. Durbin and C. G. Speziale, "Realizability of second-moment closure via stochastic analysis," *J. Fluid Mech.* **280**, 395–407 (1994).
- ⁹⁵S. S. Girimaji, "A new perspective on realizability of turbulence models," *J. Fluid Mech.* **512**, 191–210 (2004).
- ⁹⁶K. Inagaki, T. Ariki, and F. Hamba, "Higher-order realizable algebraic Reynolds stress modeling based on the square root tensor," *Phys. Rev. Fluids* **4**, 114601 (2019).
- ⁹⁷U. Schumann, "Realizability of Reynolds-stress turbulence models," *Phys. Fluids* **20**, 721–725 (1977).
- ⁹⁸T. B. Gatski and C. G. Speziale, "On the explicit algebraic stress models for complex turbulent flows," *J. Fluid Mech.* **254**, 59–78 (1993).
- ⁹⁹M. M. Rahman, P. Rautaheimo, and T. Siikonen, "Modifications for an explicit algebraic stress model," *Int. J. Numer. Methods Fluids* **35**, 221–245 (2001).
- ¹⁰⁰K. Iwamoto, Y. Suzuki, and N. Kasagi, "Reynolds number effect on wall turbulence: Toward effective feedback control," *Int. J. Heat Fluid Flow* **23**, 678–689 (2002).
- ¹⁰¹S. Hoyas and J. Jiménez, "Reynolds number effects on the Reynolds-stress budgets in turbulent channels," *Phys. Fluids* **20**, 101511 (2008).
- ¹⁰²S. Hoyas and J. Jiménez, "Scaling of the velocity fluctuations in turbulent channels up to $Re_\tau = 2003$," *Phys. Fluids* **18**, 011702 (2006).
- ¹⁰³R. Vinuesa, A. Noorani, A. Lozano-Duran, G. K. El Khoury, P. Schlatter, P. F. Fischer, and H. M. Nagib, "Aspect ratio effects in turbulent duct flows studied through direct numerical simulation," *J. Turbul.* **15**, 677–706 (2014).
- ¹⁰⁴R. Vinuesa, P. Schlatter, and H. M. Nagib, "On minimum aspect ratio for duct flow facilities and the role of side walls in generating secondary flows," *J. Turbul.* **16**, 588–606 (2015).
- ¹⁰⁵R. Vinuesa, P. Schlatter, and H. Nagib, "Secondary flow in turbulent ducts with increasing aspect ratio," *Phys. Rev. Fluids* **3**, 054606 (2018).
- ¹⁰⁶D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA, Philadelphia, USA, 2007), pp. 1027–1035.
- ¹⁰⁷X. Chen, F. Hussain, and Z. S. She, "Non-universal scaling transition of momentum cascade in wall turbulence," *J. Fluid Mech.* **871**, R2 (2019).
- ¹⁰⁸K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision (ECCV, Amsterdam, Netherlands, 2016)*, pp. 630–645.
- ¹⁰⁹B. Chang, L. Meng, E. Haber, F. Tung, and D. Begert, "Multi-level residual networks from dynamical systems view," in *6th International Conference on Learning Representations (ICLR, New York, USA, 2018)*.
- ¹¹⁰D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representation (ICLR, San Diego, USA, 2015)*.
- ¹¹¹Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* **521**, 436–444 (2015).
- ¹¹²D. Hendrycks and K. Gimpel, "Gaussian error linear units," *arXiv:1606.08415* (2016).
- ¹¹³V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML, Haifa, Israel, 2010)*.
- ¹¹⁴D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units," in *4th International Conference on Learning Representation (ICLR, San Juan, Spain, 2016)*.
- ¹¹⁵P. Bradshaw, "Turbulent secondary flows," *Annu. Rev. Fluid Mech.* **19**, 53–74 (1987).
- ¹¹⁶K. Musgrave, S. Belongie, and S.-N. Lim, "A metric learning reality check," *arXiv:2003.08505* (2020).
- ¹¹⁷J. Wu, H. Xiao, R. Sun, and Q. Wang, "Reynolds-averaged Navier–Stokes equations with explicit data-driven Reynolds stress closure can be ill-conditioned," *J. Fluid Mech.* **869**, 553–586 (2019).
- ¹¹⁸H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Comput. Phys.* **12**, 620–631 (1998).
- ¹¹⁹B. P. Brener, M. A. Cruz, R. L. Thompson, and R. P. Anjos, "Conditioning and accurate solutions of Reynolds average Navier–Stokes equations with data-driven turbulence closures," *J. Fluid Mech.* **915**, A110 (2021).
- ¹²⁰C. Cambon and J. F. Scott, "Linear and nonlinear models of anisotropic turbulence," *Annu. Rev. Fluid Mech.* **31**, 1–53 (1999).
- ¹²¹A. A. Mishra and S. S. Girimaji, "Toward approximating non-local dynamics in single-point pressure-strain correlation closures," *J. Fluid Mech.* **811**, 168–188 (2017).
- ¹²²P. M. Bevilacqua and P. S. Lykoudis, "Turbulence memory in self-preserving wakes," *J. Fluid Mech.* **89**, 589–606 (1978).
- ¹²³R. H. Kraichnan, "Lagrangian-history closure approximation for turbulence," *Phys. Fluids* **8**, 575–598 (1965).
- ¹²⁴R. H. Kraichnan, "Direct-interaction approximation for shear and thermally driven turbulence," *Phys. Fluids* **7**, 1048–1062 (1964).
- ¹²⁵P. E. Hamlington and W. J. A. Dahm, "Reynolds stress closure including non-local and nonequilibrium effects in turbulent flows," in *39th AIAA Fluid Dynamics Conference* (AIAA, 2009), p. 4162.
- ¹²⁶P. Y. Chou, "On velocity correlations and the solutions of the equations of turbulent fluctuation," *Q. Appl. Math.* **3**, 38–54 (1945).
- ¹²⁷C. G. Speziale, "On nonlinear k-l and k- ϵ models of turbulence," *J. Fluid Mech.* **178**, 459–475 (1987).
- ¹²⁸D. B. Taulbee, "An improved algebraic Reynolds stress model and corresponding nonlinear stress model," *Phys. Fluids* **4**, 2555–2561 (1992).
- ¹²⁹P. E. Hamlington and W. J. A. Dahm, "Reynolds stress closure for nonequilibrium effects in turbulent flows," *Phys. Fluids* **20**, 115101 (2008).