

Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence

Karthik Duraisamy

Department of Aerospace Engineering, University of Michigan, Ann Arbor, Michigan 48109, USA



(Received 14 September 2020; accepted 6 January 2021; published 12 May 2021)

This work presents a review and perspectives on recent developments in the use of machine learning (ML) to augment Reynolds-averaged Navier-Stokes (RANS) and large eddy simulation (LES) models of turbulent flows. Different approaches of applying supervised learning to represent unclosed terms, model discrepancies, and subfilter scales are discussed in the context of RANS and LES modeling. Particular emphasis is placed on the impact of the training procedure on the consistency of ML augmentations with the underlying physical model. Techniques to promote model-consistent training, and to avoid the requirement of full fields of direct numerical simulation data are detailed. This is followed by a discussion of physics-informed and mathematical considerations on the choice of the feature space, and imposition of constraints on the ML model. With a view towards developing generalizable ML-augmented RANS and LES models, outstanding challenges are discussed, and perspectives are provided. While the promise of ML-augmented turbulence modeling is clear, and successes have been demonstrated in isolated scenarios, a general consensus of this paper is that truly generalizable models require model-consistent training with careful characterization of underlying assumptions and imposition of physically and mathematically informed priors and constraints to account for the inevitable shortage of data relevant to predictions of interest. Thus, machine learning should be viewed as one tool in the turbulence modeler's toolkit. This modeling endeavor requires multidisciplinary advances, and thus the target audience for this paper is the fluid mechanics community, as well as the computational science and machine learning communities.

DOI: [10.1103/PhysRevFluids.6.050504](https://doi.org/10.1103/PhysRevFluids.6.050504)

I. INTRODUCTION

Over the past decade, with the availability of rich datasets from direct numerical simulations (DNS) and experiments, there has been a rapid growth in the use of machine learning methods in fluid mechanics [1]. The emergence of data science as a discipline in its own right, and broad accessibility of essential machine learning tools has fostered these advances. While rapid progress has been made in *data-driven* methods in areas such as flow decomposition [2], governing equation discovery [3] (for simple systems), and reinforcement learning [4] for flow control, predictive modeling of turbulent flow has followed a *data-enabled* paradigm. In other words, existing modeling knowledge is leveraged, and data is used to *improve* existing model structures. This is mainly because of the requirement of simulation models to be applicable across different flow regimes, geometric configurations, and the need to incorporate boundary conditions, all of which necessitate a coarse-grained partial differential equation setting. This paper is thus specifically focused on recent developments in the use of machine learning to *augment* Reynolds-averaged Navier-Stokes (RANS) and large eddy simulation (LES) models. While it is well argued that data has always played an important role in the development of models of turbulent flows, the past

decade has witnessed attempts towards a more comprehensive and formal use of data in model development [5].

The earliest application of formal machine learning in fluid mechanics is hard to ascertain. The application to particle tracking by Teo *et al.* [6] and the application to near-wall reconstruction by Milano *et al.* [7] appear to be among the first explorations in experimental and computational fluid dynamics, respectively. In the context of predictive turbulence modeling, the use of ML has its origins in the uptick in research activities in uncertainty quantification in physical modeling 15 years ago. A comprehensive review of uncertainty quantification (UQ) in RANS modeling can be found in Ref. [8]. A notable UQ work from this era, and one that is relevant to the present context is that of Oliver and Moser [9], who introduced a discrepancy to the Reynolds stress predicted by a baseline RANS model, in the form of a spatially dependent Gaussian random field. Dow and Wang [10] applied a similar approach, augmenting the eddy viscosity rather than the Reynolds stress. In contrast to the representation of Oliver and Moser, however, Dow and Wang assimilated data from multiple geometries rather than from a single geometry. Though these papers do not involve ML, they are particularly relevant to the emergence of ML applications, because they address model inadequacies in a nonparametric fashion, in contrast to earlier work which was focused on parameter calibration.

The first wave of ML applications in RANS modeling can be tracked to Tracey *et al.* [11]. In this work, supervised learning was performed in an *a priori* setting on DNS data, with the goal of reconstructing discrepancies in the Reynolds stress anisotropy tensor. The key advance from the UQ work was the transformation of model discrepancies from the spatial domain x to a feature space $\eta(x)$, consisting of quantities such as mean velocity gradients. Vollant *et al.* [12] applied similar feature space learning in an optimal estimation setting for subgrid scale modeling for LES. Further advances in enforcing consistency between the data and the model [13,14], embedded invariance [15], symbolic regression [16], etc., characterize the first phase of research. In other early work in the field, Ma *et al.* [17,18] used neural networks to model the interphase mass and momentum fluxes in multiphase flow simulations. Since 2016, this field has exploded in activity as a large number of researchers have pursued ML augmentation and applications have expanded to LES, multiphase flows, and combustion modeling.

Section II establishes the problem statement of coarse-graining and closure modeling, and is presented in a form that is also accessible to the computational science community. Section III presents the ways in which model discrepancies have been represented via supervised learning algorithms. This is the area that has seen the highest level of activity in the literature. Section IV studies the process by which the models are trained. Particularly, the concept and importance of achieving consistency between the learning and prediction environments is emphasized. Section V details general principles relevant to the selection of an appropriate feature space. Section VI discusses ways in which physics-informed constraints can be placed on the ML model and ML-augmented models. Section VII presents additional challenges and perspectives towards the goal of developing reliable and robust ML-augmented models for turbulence.

II. PROBLEM STATEMENT

Represent the highest fidelity simulation (e.g., Navier-Stokes equations, as solved by a direct numerical simulation) in the form

$$\mathcal{R}(q) = 0, \tag{1}$$

where \mathcal{R} is a partial differential equation (PDE) operator, and q are the state variables (pressure, velocity, energy, etc.). In practical problems, such as in an airplane wing or a gas turbine combustor, the level of spatial and temporal resolution required is so high that direct simulations will be unaffordable for decades to come. Thus, reduced-fidelity models are sought. In reduced-fidelity modeling using RANS and LES, the state variables are decomposed into coarse-grained variables \tilde{q} and unresolved variables $\hat{q} = q - \tilde{q}$. For example, in LES, the coarse-graining operation may

TABLE I. Examples of closure terms and related quantities. u_m, T_m : velocity, temperature; k_m, ϵ_m : turbulent kinetic energy and dissipation rate; S_m : strain rate; Δ : filter size.

Closure term	F	\tilde{q}_m	\tilde{s}_m	$\tilde{\eta}_m$
Reynolds stress (RANS)	$\tau = \widetilde{(\hat{u} \otimes \hat{u})}$	\tilde{u}_m	k_m, ϵ_m	$\tilde{S}_m \frac{k_m}{\epsilon_m}$
Subgrid scalar flux (LES)	$f = \widetilde{uT} - \tilde{u}\tilde{T}$	\tilde{u}_m, \tilde{T}_m	—	$\Delta^2 \tilde{S}_m \nabla \tilde{T}$

represent implicit or explicit low-pass filtering, and in RANS, it may represent ensemble averaging. Applying such a coarse-graining operation to $\mathcal{R}(q)$ yields

$$\widetilde{\mathcal{R}(q)} = 0 \rightarrow \mathcal{R}(\tilde{q}) + \mathcal{N}[F(q)] = 0. \quad (2)$$

It is notable that Eq. (2) is *not* an approximation. While the form of the operator \mathcal{N} is known, and the quantity F is well defined, the latter is *not closed* in the coarse-grained variables \tilde{q} . For instance, in incompressible single phase flow, the Navier-Stokes equations are given by

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = -\frac{1}{\rho} \nabla p + \nu \nabla^2 u,$$

where the unknowns are $q \equiv \{u, p\}$ (the velocity and pressure), and ν, ρ represents the kinematic viscosity and density. When ensemble averaging $\tilde{\cdot}$ is introduced and applied to the above equation, we get

$$\frac{\partial \tilde{u}}{\partial t} + \tilde{u} \cdot \nabla \tilde{u} = -\frac{1}{\rho} \nabla \tilde{p} + \nu \nabla^2 \tilde{u} - \nabla \cdot (\widetilde{\hat{u} \otimes \hat{u}}).$$

These are the RANS equations and thus, $\mathcal{N}(\cdot) \equiv -\nabla \cdot (\cdot)$ and $F(q) \equiv \widetilde{\hat{q} \otimes \hat{q}}$. In this work, the quantity F is used to generically represent unclosed terms such as the Reynolds stresses, subgrid scale stresses, scalar fluxes, turbulence-chemistry interactions, etc.

The goal of turbulence modeling is to overcome the closure problem in Eq. (2) by defining a RANS or LES model in the form

$$\mathcal{R}(\tilde{q}_m) + \mathcal{N}[F_m(\tilde{q}_m, \tilde{s}_m)] = 0. \quad (3)$$

The main objective in turbulence modeling is to construct a closure approximation $F_m \approx F$ in terms of the modeled coarse-grained variables \tilde{q}_m and some secondary variables \tilde{s}_m , which may themselves involve additional transport equations $\mathcal{G}_m(\tilde{s}_m, \tilde{q}_m) = 0$. For example, \tilde{q}_m can represent the modeled ensemble-averaged velocity and \tilde{s}_m can represent the modeled dissipation rate of turbulent kinetic energy as shown in Table I.

The construction of F_m and \mathcal{G}_m is a highly intricate process, evolving over several decades through a combination of physical insight, mathematics, and empiricism [19]. While data has always been an enabler of closure modeling, the ways in which data has been used thus far has perhaps not been comprehensive. Partly as a consequence of this, many closure modeling approaches have saturated in accuracy [20,21] over the past decade (or more).

In this review, we specifically discuss the use of ML in augmenting coarse-grained RANS and LES models, and will use \tilde{q}, \tilde{s} to denote *coarse-grained* quantities derived from a DNS or experiment, and \tilde{q}_m, \tilde{s}_m will be modeled representations of the same quantities. $Y(\tilde{q}), Y(\tilde{q}_m)$ will be used to represent observables of the resolved field (e.g., skin-friction coefficient on a wall). The machine learning model will be generically represented by $\delta_m(\tilde{\eta}_m; w)$, where $\tilde{\eta}_m$ are features (inputs) and w are the parameters of the learning representation (e.g., weights and biases of a neural network). For illustrative purposes, examples of the above quantities are shown in Table I. The machine learning representation δ_m can represent the entire closure term (for instance, Beck *et al.* [22] develop an ML model for the subgrid scale stresses in LES), or part of a closure term (for

instance, Schmelzer *et al.* [23] extract an ML model as a correction term to an eddy viscosity model of the Reynolds stress tensor in RANS).

III. REPRESENTATION

Much of the research in ML-augmented turbulence modeling has gone towards different ways of representing model inadequacy and embedding them appropriately in RANS and LES models. It is not the intent of this section to cover all of the work in this area, but rather an attempt is made at presenting a prototypical set of representations. This section is divided into two parts: The first discusses techniques to model the impact of the unresolved physics on the coarse-grained variables $F_m(\tilde{q}_m, \tilde{s}_m)$, and the second part discusses techniques to model the unresolved quantities \hat{q}_m directly.

A. Closure term and model inadequacy representations

Tracey *et al.* [11] represented the Reynolds stress tensor as

$$\tau_m = 2k_m \left[\frac{1}{3}I + \tilde{V}_m (\tilde{\Lambda}_m + \delta_m(\tilde{\eta}_m; w) \tilde{V}_m^T) \right],$$

where \tilde{k}_m is the modeled turbulent kinetic energy and $\tilde{V}_m, \tilde{\Lambda}_m$ are matrices that contain the eigenvectors and eigenvalues of the modeled Reynolds stress anisotropy, respectively. $\delta_m(\tilde{\eta}_m; w)$ is a kernel regression-based ML model that models the error in anisotropy via supervised learning on a DNS dataset. The features $\tilde{\eta}_m$ include the eigenvalues of the anisotropy tensor, the ratio of the production-to-dissipation rate (of turbulent kinetic energy), and a marker function that masks regions of thin shear layers from augmentation. In a series of papers [24,25], Xiao and co-workers expanded the description above to include a more comprehensive perturbation of not just the eigenvalues, but also the eigenvectors and the turbulent kinetic energy, and considered a broad range of features. Duraisamy *et al.* [13,26,27] applied similar feature-based augmentations to transport equations, rather than directly to the Reynolds stress.

Ling *et al.* [28] proposed a neural network architecture to learn the coefficients of a tensor basis expansion for the Reynolds stresses in the form

$$\tau_m = 2k_m \left(\frac{1}{3}I + \left[\sum_{n=1}^{10} \delta_m^{(n)}(\tilde{\eta}_m; w) T^{(n)}(\tilde{\mathcal{S}}_m, \tilde{\mathcal{Q}}_m) \right] \right), \quad (4)$$

where $T^{(n)}(\tilde{\mathcal{S}}_m, \tilde{\mathcal{Q}}_m)$ are the tensorial basis [29] (with respect to the strain rate and vorticity tensors) and $\delta_m^{(n)}(\tilde{\eta}_m; w)$ are the coefficients which are represented using neural networks. It is notable that the features $\tilde{\eta}_m$ are taken to be five invariants based on $\tilde{\mathcal{S}}_m$ and $\tilde{\mathcal{Q}}_m$, and are objective by definition. This so-called tensor basis neural network has since been extended in application to model turbulent heat fluxes [30] and scalar fluxes [31]. Kaandorp and Dwight [32] leverage the above integrity basis, but use random forest regression instead of neural networks.

Approaches that are similar in spirit, from the viewpoint of constructing models for the Reynolds stress anisotropy in terms of invariants of the velocity gradient tensor, are pursued in [16,33] using symbolic regression with genetic programming. Symbolic regression has the appeal of explicit and interpretable model forms that are more amenable for analysis. Reference [34] extends the above invariant-based gene-expression approach to represent turbulent diffusivity.

Schmelzer *et al.* [23] employ sparse linear regression (rather than neural networks or symbolic regression) on a library of candidate functions which are written as tensor polynomials of the aforementioned invariants. Beetham and Capecehatro [35] also follow a similar approach and further extend sparse linear regression to model drag production, drag exchange, pressure strain, and viscous dissipation in RANS of multiphase flows [36]. Sparsity-enforcing regularizers in these approaches lead to the elimination of some model terms, thus simplifying the model form.

Sarghini *et al.* [37] introduced the idea of using neural networks to aid subgrid scale modeling in LES. The feature space included the resolved velocity gradients and stresses, and the output was a

Smagorinsky-type viscosity coefficient. More recently, several researchers (e.g., [38–40]) have used similar approaches to directly relate the subgrid scale stress tensor to a feature space which includes the resolved velocity gradient tensor and mesh resolution. Notable work was performed by Vollant *et al.* [12,41], who used optimal estimation theory to separate subgrid scale modeling error into parametric and model form components and neural networks to address each of the components in isolation.

Beck *et al.* [22] (using convolutional neural networks on the resolved field) and Xie *et al.* [42,43] (using the resolved local flow gradients) directly model the subgrid scale forcing $f \approx \nabla \cdot \tau_m$, since small errors in the subgrid scale *stresses* can otherwise generate large errors in the momentum equation.

Over the past year, similar approaches to subgrid scale modeling have also been applied to reacting flows and multiphase flows. As examples, Ref. [44] uses convolutional neural networks on the full field of the resolved progress variable to construct subgrid flame density function estimates; and Ref. [45] uses neural networks to learn the filtered progress variable source term.

B. Subgrid scale representations

The idea of approximate deconvolution as a path towards closure modeling was introduced by Stolz and Adams [46]. The basic idea revolves around reconstructing subfilter contributions \hat{q}_m from filtered quantities \tilde{q}_m . Once such an approximation is constructed, the subgrid scale stresses can be approximated as $\tau_m = \hat{q}_m \otimes \tilde{q}_m - q_m \otimes q_m$. While the original idea of approximate deconvolution was based on analytically inverting a known filtering kernel, [47,48] employ a neural network to represent the deconvolution $\hat{q}_m = \delta_m(\tilde{q}_m; w)$, based on a localized stencil of inputs. Note that unlike in the classical approximate deconvolution strategy, the filtering kernel is not assumed *a priori*.

A related idea to deconvolution is super-resolution. While deconvolution is aimed at modeling subfilter quantities on the same discretization as the resolved quantities, super-resolution targets the extraction of a finer resolution field; for instance, extracting finer mesh quantities or a higher-order accurate solution. Inspired by advances in imaging, [49,50] introduced the idea of super-resolution to fluid mechanics by leveraging neural networks to represent the mapping from the coarse field to the fine field. Reference [51] extended applications to the spatiotemporal setting, where given a coarse field at two time instances t and $t + \Delta t$, super-resolved spatiotemporal fields are generated at many smaller time instances in the interval $[t \dots t + \Delta t]$. References [52] and [53] use variants of generative adversarial networks (GAN) for super-resolution. Though these models have mostly been evaluated in an *a priori* sense, these approaches are highly expressive, and present promise for online modeling.

In Ref. [54], a GAN is trained to super-resolve a coarse solution (filtered DNS) to a fine solution (DNS). The resolution (image pixels) of input and output are kept fixed. The model is used to obtain fine scales from the coarse LES solution, which is used to compute the subgrid terms (in the momentum and scalar equations).

DNS of dispersed multiphase flow is expensive since the field around each particle needs to be resolved leading to a large number of variables. Reference [55] uses a GAN to generate the velocity field around a group of particles in a small section of the domain locally, which is then applied patchwise over the entire domain.

IV. TRAINING AND CONSISTENCY

The previous section was focused on the *representation* of the unclosed terms, i.e., on how to specify the model form of the closure. In this section, we study the process by which these models are extracted from data. The concept of consistency between the learning and prediction environments is explored. To simplify notations, we will use $\delta_m(\tilde{\eta}_m; w)$ to represent the ML models, and $\mathcal{R}_a[\hat{q}_m, \tilde{s}_m, \delta_m(\tilde{\eta}_m; w)] = 0$ to represent all of the transport equations (e.g., Reynolds-averaged mass, momentum, energy equations, and transport equations for scale-providing variables) of the

machine learning augmented model. Such a definition is used to exemplify the representations in Sec. III in a compact manner.

A. *A priori* training

In a majority of the references above (e.g., [11,15,16,25,36]), training is performed by directly extracting δ (the target of the learning model) and $\tilde{\eta}$ (the input to the learning model) from the DNS. Following this, one posits an ML model $\delta_m(\cdot; w)$, and the following supervised learning problem is posed:

$$\min_w \mathcal{L}[\delta, \delta_m(\tilde{\eta}; w)], \quad (5)$$

where \mathcal{L} is a generic loss function that is a proxy for frequentist or Bayesian inference. For instance, in sparse regression,

$$\mathcal{L}[\delta, \delta_m(\tilde{\eta}; w)] \equiv \|\delta - \delta_m(\tilde{\eta}; w)\|_2^2 + \lambda \|w\|_1.$$

After the training (and cross-validation) process is complete, the trained model $\delta_m(\cdot; w)$ is embedded in a baseline model in a predictive setting:

$$\mathcal{R}_a[\tilde{q}_m, \tilde{s}_m, \delta_m(\tilde{\eta}_m; w)] = 0. \quad (6)$$

This training approach is natural, nonintrusive (i.e., the solver is not involved), and provides opportunities to directly impose physics-based constraints. However, consistency with the model can become a critical issue as has been pointed out in the context of RANS [5,56,57] and LES [58]. In short, *a priori* training establishes the consistency of the ML model with the DNS field, but does not guarantee consistency with the RANS or LES environment. The following are some of the main reasons for the loss of consistency:

Feature mismatch between training and prediction. During the training process [Eq. (5)], the coarse-grained features $\tilde{\eta}$ from the DNS are used as inputs to the learning model, whereas in the prediction process, coarse-grained model features $\tilde{\eta}_m$ are used. Thus, for this approach to work well, the model has to predict the features $\tilde{\eta}_m$ very accurately (i.e., $\tilde{\eta}_m = \tilde{\eta}$). This is typically difficult to achieve because the features may depend on secondary variables which may not be predicted well by the models. For instance, a feature \tilde{S}_ϵ^k may not equal $\tilde{S}_m^{k_m}$ because k_m, ϵ_m can be very different from k, ϵ even if \tilde{S}_m is close to \tilde{S} . This is because models are typically constructed to predict certain quantities (e.g., mean flow, Reynolds shear stress, etc.) well, but not all quantities. This is especially true in RANS where several secondary quantities are used to provide length and time scales which should not be interpreted to have a one-on-one correspondence with the DNS quantities.

Accumulation of errors in prediction. Consider a super-resolution operation of going to a fine field from a coarse field via a neural network. The results of such a process are shown in Fig. 1. The neural network is trained (offline) to represent $u(t) = f_{NN}[\tilde{u}(t); w]$, whereas when it is embedded in the solver, it predicts $u_m(t) = f_{NN}[\tilde{u}_m(t); w] = f_{NN}[\tilde{u}(t) + e(t); w]$. As the error $e(t)$ accumulates over time, the neural network is required to make predictions based on a field that is corrupted by error, which becomes futile as seen by the lack of fine-scale structure at $t^* = 6.0$ in the online prediction. Such errors are typical of most practical LES computations because even though one attempts to model only the unresolved scales, the scales that are barely larger than the filter size are often very poorly resolved. Thus, the learning model has to consider the corruption of the resolved scales.

Balance between model terms. It is well recognized in the turbulence modeling community that successful *a priori* evaluation is neither a necessary (e.g., Smagorinsky model) nor a sufficient condition for successful predictive models. Even second moment closure-based RANS models are formulated such that the balance between model terms is managed in a manner that ensures a degree of success in predictive outcomes. Reference [59] calculated the working variables in turbulence

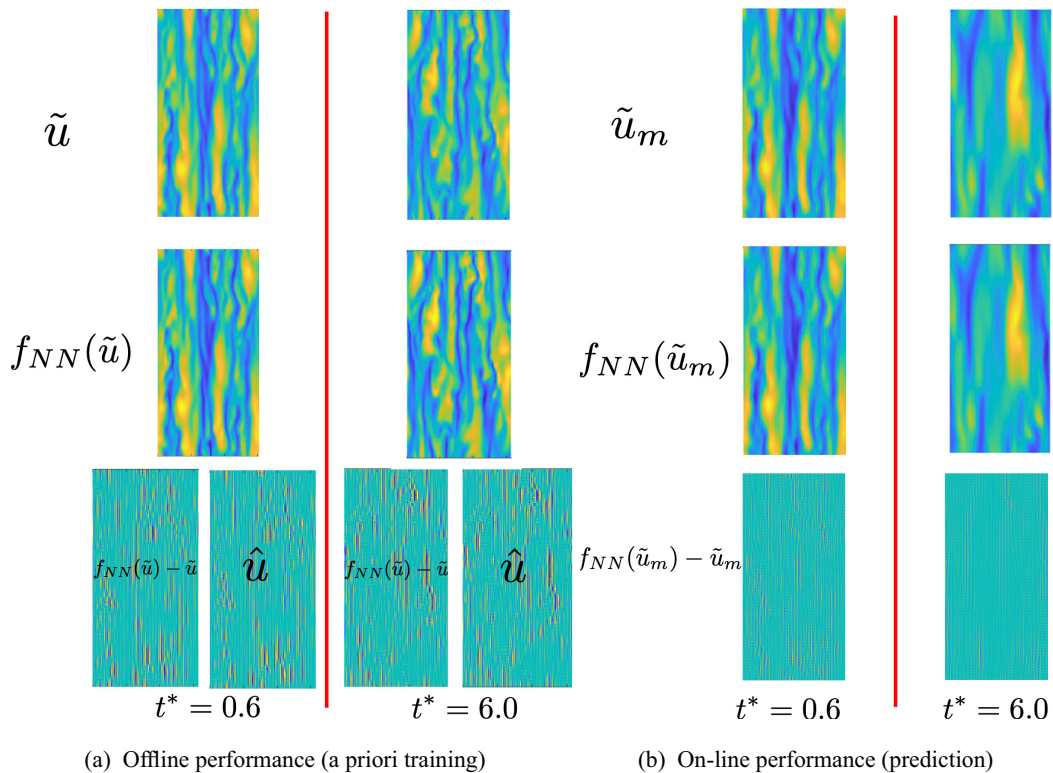


FIG. 1. Impact of error accumulation in online simulations. Top row: coarse-grained field; middle row: fine field; bottom row: fine scales.

models using DNS and LES datasets for one- and two-equation models and concluded that the use of exact values of the turbulent kinetic energy and dissipation rate in the modeled eddy viscosity did not improve its performance. Reference [60] showed that even substituting Reynolds stress fields from reputable DNS databases may not lead to satisfactory velocity fields. Further, [61] investigated potential conditioning problems that arise when explicitly trained ML models are injected into existing turbulence models.

Another major impediment to the use of *a priori* techniques is that a *full field of DNS data is required* to train the model. Since DNS data will not be available in practical regimes, this is a major limitation.

In spite of the above challenges, good results have been reported in the literature. For the purposes of generalization, it will be a good practice to ascertain the correlations between $\tilde{\eta}$ and $\tilde{\eta}_m$ (beyond δ and δ_m) such that the degree of loss in consistency can be monitored.

B. Model-consistent training

The previous discussion highlights the importance of learning model augmentations in an (imperfect) modeled environment rather than the (perfect) environment of the coarse-grained DNS field. Enforcing consistency, however, requires the solution of inverse problems that minimize a discrepancy between the model output and the data. Thus, the model $R_a(\cdot)$ is involved in the training process in contrast to *a priori* training.

Given a sparse dataset \tilde{Y}^i from experiments or DNS, the so-called field inversion approaches [10,13,62] seek the spatiotemporal model augmentation field $\delta_m^i(x, t)$ via the following

PDE-constrained inverse problem:

$$\min_{\delta_m^i} \mathcal{L}[Y^i, Y_m^i(\delta_m^i)], \quad \text{s.t. } \mathcal{R}_a[\tilde{q}_m, \tilde{s}_m, \delta_m(\tilde{\eta}_m; w)] = 0. \quad (7)$$

Such an augmentation is consistent with the underlying model, and unlike the *a priori* approach which requires δ^i (rather than Y^i) from a detailed DNS field, this approach, in principle, can assimilate information from available and potentially sparse data. In practice, one has to ensure that the data Y^i is informative of the underlying model discrepancy.

The field inversion (FI) approach has been pursued in the context of augmentation of eddy viscosity [10], transport equation terms [62–64], Reynolds stresses [65,66], and in the mean flow momentum equation [67]. Given a dataset Y^i , the FI approach yields δ_m^i , which is a spatiotemporal field, and is thus problem specific. To convert this into a generalizable augmentation, FI is performed on k datasets that are presumably informative of the model discrepancy, following which a set of targets $\delta_m^* = \{\delta_m^1, \dots, \delta_m^k\}$ and features $\tilde{\eta}_m^* = \{\tilde{\eta}_m^1, \dots, \tilde{\eta}_m^k\}$ are collected. Supervised learning can then be performed:

$$\min_w \mathcal{L}[\delta_m^*, \delta_m(\tilde{\eta}_m^*; w)]. \quad (8)$$

In contrast to Eq. (5) in which coarse-grained DNS data is used to train the model, Eq. (8) creates data from the model outputs to train a generalized augmentation. This combination of FI and ML as given in Eqs. (7) and (8) has been pursued by several researchers [14,27,56,63,64,68] in various applications in turbulence and transition modeling.

While the aforementioned FIML approach promotes consistency between the model outputs and the learning environment, imperfections in the learning process [Eq. (8)] can introduce a degree of inconsistency between the learning and prediction environments. This remaining inconsistency can be eliminated by integrating the inference step with the learning step in the form

$$\min_w \mathcal{L}\{Y, Y_m[\delta(\tilde{\eta}_m; w)]\}, \quad \text{s.t. } \mathcal{R}_a[\tilde{q}_m, \tilde{s}_m, \delta_m(\tilde{\eta}_m; w)] = 0. \quad (9)$$

In this approach, the ML output is first fed into the model, and the model output is used in the loss function. Notably, both of these steps are integrated within a single optimization step that updates the weights of the ML model. This tightly coupled approach has recently been employed in the context of RANS [69,70] and LES [58], and ensures full consistency between the learning and prediction environments. Figure 2 exemplifies the importance of consistent training.

It has to be mentioned that one minor drawback of the integrated approach [Eq. (9)] is that features have to be selected before the inference, whereas separation of the FI [Eq. (7)] and ML [Eq. (8)] stages yields more flexibility.

It is notable that model-consistent training involves large scale inverse problems, typically necessitating the use of adjoint-driven [71] optimization techniques [14,58,70]. Given the intrusive nature of adjoints, this presents major challenges to development, and has proven to be a barrier for researchers to develop model-consistent ML augmentations. This is especially a challenge in tightly coupled approaches, as the optimization considers the scale and nonlinearities of the PDE and highly nonconvex loss function landscapes of neural networks. An additional challenge in LES is that chaoticity can lead to unstable adjoint solutions that require special treatment [72,73].

To circumvent the complexity of the aforementioned tightly coupled optimization problems, weakly coupled techniques such as “embedded learning” [69], “iterative machine learning” [74], “CFD-driven machine learning” [75], and “closed loop training” [57] have been proposed. While some of these approaches still require a full field of DNS data as in *a priori* training, these techniques represent a movement towards establishing model consistency without the need for complex adjoint-driven machinery, thus reducing barriers to development of ML-based turbulence models.

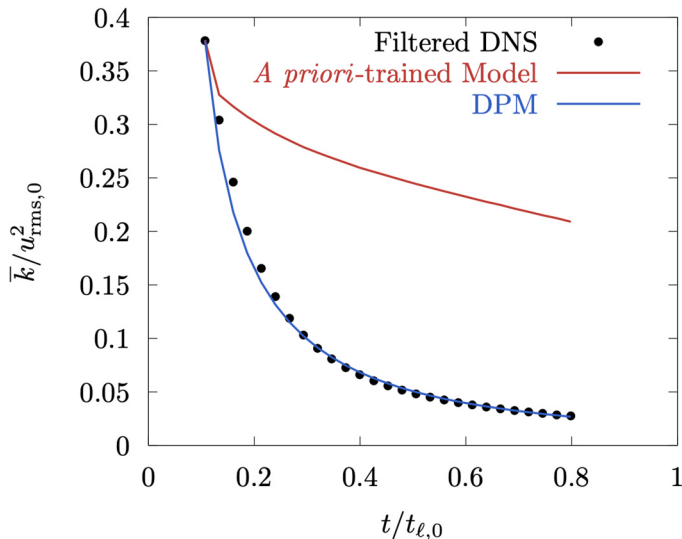


FIG. 2. Comparison of *a priori*-trained subgrid scale closure and model-consistent learning/inference (DPM) applied to LES of homogeneous isotropic turbulence (Ref. [58]).

Convergence characteristics of all the methods mentioned in this section have to be studied in a mathematically rigorous fashion, and will be a fruitful research direction. The choice of the loss function is also a topic that has not been studied in earnest.

V. FEATURE SELECTION

The ML models discussed above are based on regression and are meant to be interpolative in feature space η_m . However, if features are properly selected, and the feature space is adequately populated, embedding the output of these regression models within a suitable physics-based model $\mathcal{R}_a[\tilde{q}_m, \tilde{s}_m, \delta_m(\tilde{\eta}_m; w)] = 0$ can yield better predictive properties on unseen geometries and flow configurations.

In the limit of the availability of an infinite amount of data, feature selection can be posed as an unsupervised learning problem. In a practical turbulence modeling scenario, however, selection of features should be guided by turbulence modeling principles [25] as well as by the amount and type of available data. The following are general guidelines based on prevailing practices:

Local nondimensionalization. To ensure applicability across different problems, the features should be locally nondimensionalized [15,76]. For instance, if the strain-rate tensor \tilde{S} is used as a feature, nondimensionalization with respect to a local time scale (e.g., $\tilde{S}k/\epsilon$) offers a greater possibility to generalize across different configurations, when compared to global nondimensionalization.

Invariance considerations. As in traditional turbulence modeling [20], ideal features should satisfy rotational, reflectional, and frame-invariance properties. This aspect is discussed in detail in [15,25]. This should apply to both the selected features and variables used for local nondimensionalization.

Local vs nonlocal features. From the viewpoints of generalizability and implementation, it is desirable to have local features (such as $\tilde{S}k/\epsilon$). In practice, wall-distance [24,25] and wall-stress-based measures [27] appear to be important. Pressure-gradient-based features [25] have also been used as a surrogate for nonlocal information.

Data considerations. Feature selection is a well-studied topic in the machine learning community and a comprehensive summary can be found in Ref. [77]. Wang *et al.* [24] use random forest

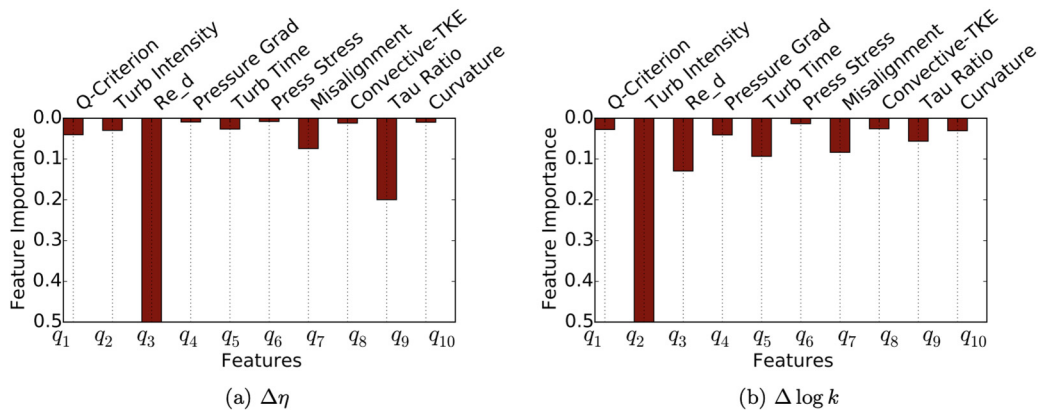


FIG. 3. Ranking important features using two different representations of model discrepancy (Ref. [24]).

regression on 47 features and rank them in order of importance (Fig. 3). Such tools can be used as a guide to narrow down to a smaller set, with the caveat that limited data might bias the choice of features and impact model reliability. Indeed, constructing a model with d features requires enough data to populate a d -dimensional feature space.

VI. CONSTRAINTS

The general philosophy of combining ML models with physics-based models relies on the notion that physics information will complement the information contained in the data to yield generalizable predictions. The imposition of constraints can effectively reduce the search space of ML models to lower dimensional manifolds consistent with the physics. In the present context of data-augmented turbulence modeling, constraints can take several forms, including the following.

Input constraints on the ML model. As discussed in Sec. V, the feature space $\tilde{\eta}_m$ can be constrained to a manifold that satisfies invariance properties.

Output constraints on the ML model. The output of the learning model may also be constrained. For instance, if the output corresponds to a Reynolds stress perturbation to a baseline model (represented by a subscript b), i.e., $\tau_m = \tau_b + \delta(\tilde{\eta}_m; w)$, realizability constraints could be enforced on $\delta(\tilde{\eta}_m; w)$. Additional examples include the case in which ML outputs correspond to quantities that are reflectionally or rotationally invariant. This can be addressed, for instance, via data augmentation [15].

Constraints on the outputs of the physics model. In model-consistent training, relevant equality and inequality constraints can be imposed on observables of the output of the physics model. For instance, in a combustion modeling setting, even if the local heat release rate is not used in the loss function, the model output can be constrained to match the integrated heat release in the experiment (or DNS) during the training stage.

Constraint satisfaction via priors. In the case of Bayesian inference, physical and mathematical information can be used to enforce priors on the parameters and outputs.

Reference [57] also presents additional perspectives on compatibility and physics-based constraints.

VII. ADDITIONAL CHALLENGES AND PERSPECTIVES

Turbulence modeling is a peculiar endeavor, with competing philosophies and paradoxes [21]. For instance, it is often not clear as to what type of physical information, however elegant it may appear, is useful for predictive outcomes. In fact, Ref. [20] goes on to say that “The central

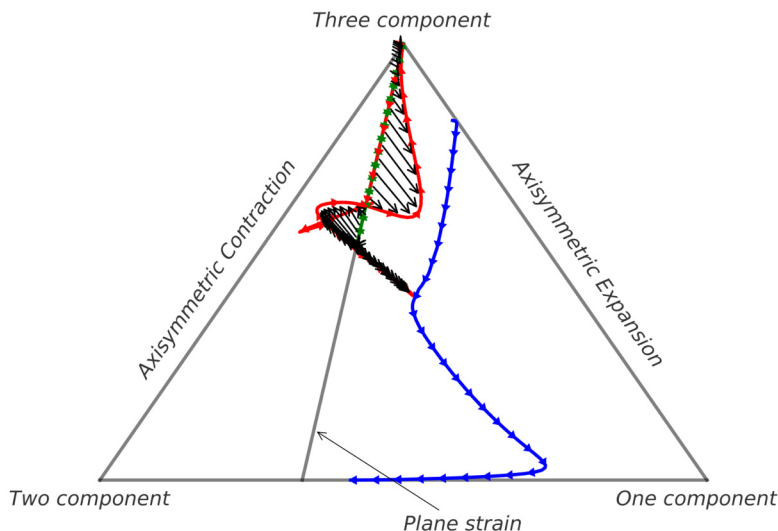


FIG. 4. Model-consistent inference based on velocity data in channel flow. Blue: anisotropy predicted by DNS; red: anisotropy predicted by the inferred model.

role of creativity and free intuition introduces a danger of *proliferation*. Any type of new term can be proposed, and many will satisfy the consensus constraints such as Galilean invariance, so that rejecting them becomes a matter of opposing intuition.” It has to be recognized that ML augmentations are being introduced to a scenario which is already extremely complex, and sometimes counterintuitive. Thus underlying intricacies and Occam’s razor must be appreciated. We begin this section by outlining additional challenges, and then offer some perspectives towards the development of generalizable ML-augmented models.

Irrecoverable model discrepancies. Coarse-grained simulations of turbulence, as in the context of RANS and LES, have to contend with irrecoverable loss of information which is a consequence of the fact that, at any given instant in time, there are infinite realizations of velocity fields that are compatible with the coarse-grained field. Thus, each of these realizations might evolve dynamically in different ways, which can only be captured in an average sense in a coarse-grained model [32,78]. Further, the modeling structure by itself (e.g., single-point closure [5], Markovian closures [79], etc.) may introduce errors not addressable by the ML augmentation.

Identifiability. As a consequence of the fact that (a) the model inadequacy cannot be fully addressed by the ML augmentation alone, and (b) the available data may not be sufficient to learn the true model form, the augmentation may not be identifiable. Figure 4 illustrates the latter aspect. A model-consistent perturbation to the Reynolds stress anisotropy tensor based on the mean velocity was found to yield highly accurate mean velocity and Reynolds shear stress predictions, but the anisotropy was not predicted accurately. This could indeed be remedied if the entire Reynolds stress tensor was used as the data. This example, however, is used as a proxy for the lack of data for practical configurations: DNS would be infeasible, and experimental methods will offer sparse measurements of certain field quantities.

Impact of discretization error and filtering. The augmentation of LES models is complicated by the fact that the resolved scales are seldom free of numerical errors, and thus disambiguating numerical errors from modeling errors is a challenge. This is especially true when explicit filtering [80,81] is not employed. Thus, ascertaining what exactly has to be represented by the ML algorithm is typically unclear. Model-consistent training can account for the impact of numerical errors in a more precise manner, at least as far as reconstructing the training data is concerned. A further complication in LES is that the discrepancy between the filtered Navier-Stokes equations and

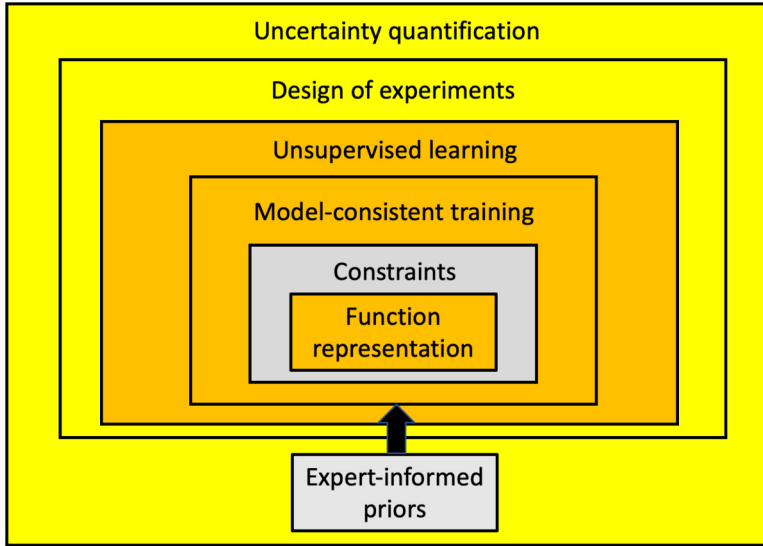


FIG. 5. Schematic representing key layers in the development of generalizable ML-augmented models.

the model equations may be difficult to assess. This is especially true when the filter is implicit, and model-consistent training as presented in Sec. IV requires further refinement. Further, as pointed out by Lund [80], even explicit filtering may not fully address these deficiencies. Much further research is required towards the end of developing techniques for generalizable ML-augmented LES models.

Interpretability. Interpretable ML models are desirable from the viewpoint of analysis, implementation, reproducibility, and wider use. Models based on symbolic regression [16,34] are appealing in this regard. Reference [82] offers a survey of recent research on the interpretability of a broader range of ML models. It should be mentioned that while interpretable models are desirable, they do not automatically become applicable beyond the scenarios in which they were trained.

As with the development of all ML-based models, careful consideration should be given to achieving predictive capabilities beyond the training set. While the topics discussed in this paper present some directions towards achieving generalization, the author's perspective is that ML augmentation should be considered as just another tool in turbulence modeling. Judicious use of supervised and unsupervised learning, grounded by physics constraints and mathematical rigor *and an understanding of the information (and lack thereof) contained in the available data* is required to derive generalizable models.

A schematic of a comprehensive approach towards developing ML-augmented models is shown in Fig. 5. Since every step of the modeling process involves assumptions, quantifying the impact of these assumptions in the form of distributions, and formally accounting for them in inverse and forward modeling will add robustness to model development. Over the past decade, uncertainty quantification formalisms have been applied to turbulence models [8,9,83]. While the treatment of parametric uncertainties is well established and is purely a matter of applying scalable computational algorithms, addressing model form uncertainties continues to present an open topic in computational science.

It is the author's opinion that the promise of ML-augmented turbulence modeling is clear. Preliminary successes have been demonstrated by small groups of individuals in a limited set of problem configurations. Several researchers have shown that ML-augmented modeling can offer improved predictions over classical models in problems that were either part of the training dataset or in problems that are related to the training set (e.g., an airfoil with a slightly different shape with respect to the training datasets [27]). The goal of more generalizable models has, however,

not been achieved. Further progress requires the community to move beyond the publication of journal articles, and establish standards/benchmarks and foster a collaborative ecosystem. ML (and artificial intelligence in general) have revolutionized certain fields rapidly, mainly because of (a) wide use of real-world benchmarks (e.g., ImageNet), (b) standardizing evaluation and accessibility of underlying tools, and (c) promoting reproducibility of research.

As a final point, throughout the history of turbulence (and combustion) modeling, the desire for deterministic models has resulted in developers presenting “unique” model parameters and model forms, after balancing many considerations. Documentation of candidate model forms that were explored during model development and specification of parameter ranges can help inform rigorous priors and greatly benefit data-driven modeling.

ACKNOWLEDGMENTS

The author acknowledges support from the NSF CBET program (Grant No. 1507928), ONR Subsurface Hydrodynamics program (Grant No. N00014-17-1-2200), and NASA TTT program (Grant No. 80NSSC18M0149). Aniruddhe Pradhan’s help with the super-resolution results is gratefully acknowledged.

- [1] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, Machine learning for fluid mechanics, *Annu. Rev. Fluid Mech.* **52**, 477 (2020).
- [2] K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. T. Dawson, and C.-A. Yeh, Modal analysis of fluid flows: Applications and outlook, *AIAA J.* **58**, 998 (2020).
- [3] S. Rudy, A. Alla, S. L. Brunton, and J. N. Kutz, Data-driven identification of parametric partial differential equations, *SIAM J. Appl. Dyn. Syst.* **18**, 643 (2019).
- [4] S. Verma, G. Novati, and P. Koumoutsakos, Efficient collective swimming by harnessing vortices through deep reinforcement learning, *Proc. Natl. Acad. Sci. U.S.A.* **115**, 5849 (2018).
- [5] K. Duraisamy, G. Iaccarino, and H. Xiao, Turbulence modeling in the age of data, *Annu. Rev. Fluid Mech.* **51**, 357 (2019).
- [6] C. Teo, K. Lim, G. Hong, and M. Yeo, A neural net approach in analyzing photograph in PIV, *Conference Proceedings 1991 IEEE International Conference on Systems, Man, and Cybernetics* (IEEE, New York, 1991), pp. 1535–1538.
- [7] M. Milano and P. Koumoutsakos, Neural network modeling for near wall turbulent flow, *J. Comput. Phys.* **182**, 1 (2002).
- [8] H. Xiao and P. Cinnella, Quantification of model uncertainty in RANS simulations: A review, *Prog. Aerospace Sci.* **108**, 1 (2019).
- [9] T. A. Oliver and R. D. Moser, Bayesian uncertainty quantification applied to RANS turbulence models, *J. Phys.: Conf. Ser.* **318**, 042032 (2011).
- [10] E. Dow and Q. Wang, Quantification of structural uncertainties in the $k-\omega$ turbulence model, *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference* (AIAA, 2011), paper 2011–1762.
- [11] B. Tracey, K. Duraisamy, and J. Alonso, Application of supervised learning to quantify uncertainties in turbulence and combustion modeling, *51st AIAA Aerospace Sciences Meeting*, paper 2013-0259, 2013.
- [12] A. Volland, G. Balarac, G. Geraci, and C. Corre, Optimal estimator and artificial neural network as efficient tools for the subgrid-scale scalar flux modeling, Tech.report, Proceedings of Summer Research Program, *Center of Turbulence Research*, Stanford University, Stanford, CA, 2014.
- [13] K. Duraisamy, Z. J. Zhang, and A. P. Singh, New approaches in turbulence and transition modeling using data-driven techniques, in *Proceedings of the 53rd AIAA Aerospace Sciences Meeting* (2015), Paper 2015-1284.
- [14] E. J. Parish and K. Duraisamy, A paradigm for data-driven predictive modeling using field inversion and machine learning, *J. Comput. Phys.* **305**, 758 (2016).

- [15] J. Ling, R. Jones, and J. Templeton, Machine learning strategies for systems with invariance properties, *J. Comput. Phys.* **318**, 22 (2016).
- [16] J. Weatheritt and R. Sandberg, A novel evolutionary algorithm applied to algebraic modifications of the RANS stress-strain relationship, *J. Comput. Phys.* **325**, 22 (2016).
- [17] M. Ma, J. Lu, and G. Tryggvason, Using statistical learning to close two-fluid multiphase flow equations for a simple bubbly system, *Phys. Fluids* **27**, 092101 (2015).
- [18] M. Ma, J. Lu, and G. Tryggvason, Using statistical learning to close two-fluid multiphase flow equations for bubbly flows in vertical channels, *Int. J. Multiphase Flow* **85**, 336 (2016).
- [19] P. A. Durbin, Some recent developments in turbulence closure modeling, *Annu. Rev. Fluid Mech.* **50**, 77 (2017).
- [20] P. R. Spalart, Philosophies and fallacies in turbulence modeling, *Prog. Aerospace Sci.* **74**, 1 (2015).
- [21] K. Duraisamy, P. R. Spalart, and C. L. Rumsey, Status, emerging ideas and future directions of turbulence modeling research in aeronautics, NASA Technical Memorandum 2017-219682, 2017.
- [22] A. Beck, D. Flad, and C.-D. Munz, Deep neural networks for data-driven LES closure models, *J. Comput. Phys.* **398**, 108910 (2019).
- [23] M. Schmelzer, R. P. Dwight, and P. Cinnella, Discovery of algebraic Reynolds-stress models using sparse symbolic regression, *Flow, Turbul. Combust.* **104**, 579 (2020).
- [24] J.-X. Wang, J.-L. Wu, and H. Xiao, Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data, *Phys. Rev. Fluids* **2**, 034603 (2017).
- [25] J.-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).
- [26] K. Duraisamy and P. Durbin, Transition modeling using data driven approaches, *Proceedings of the CTR Summer Program* (Stanford University, Stanford, CA, 2014), p. 427.
- [27] A. P. Singh, S. Medida, and K. Duraisamy, Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils, *AIAA J.* **55**, 1 (2017).
- [28] J. Ling, A. Kurzwaski, and J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, *J. Fluid Mech.* **807**, 155 (2016).
- [29] S. B. Pope, *Turbulent flows* (Cambridge University Press, Cambridge, UK, 2000).
- [30] P. M. Milani, J. Ling, and J. K. Eaton, Generalization of machine-learned turbulent heat flux models applied to film cooling flows, *J. Turbomach.* **142**, 011007 (2020).
- [31] P. M. Milani, J. Ling, and J. K. Eaton, Turbulent scalar flux in inclined jets in crossflow: Counter gradient transport and deep learning modeling, *J. Fluid Mech.* **906**, A27 (2021).
- [32] A. A. Mishra, G. Iaccarino, and K. Duraisamy, Sensitivity of flow evolution on turbulence structure, *Phys. Rev. Fluids* **1**, 052402 (2016).
- [33] J. Weatheritt and R. D. Sandberg, The development of algebraic stress models using a novel evolutionary algorithm, *Int. J. Heat Fluid Flow* **68**, 298 (2017).
- [34] A. Haghiri and R. D. Sandberg, Large eddy simulations of wall jets with coflow for the study of turbulent Prandtl number variations and data-driven modeling, *Phys. Rev. Fluids* **5**, 064501 (2020).
- [35] S. Beetham and J. Capecehatro, Formulating turbulence closures using sparse regression with embedded form invariance, *Phys. Rev. Fluids* **5**, 084611 (2020).
- [36] S. Beetham, R. O. Fox, and J. Capecehatro, Sparse identification of multiphase turbulence closures for coupled fluid-particle flows, [arXiv:2004.08997](https://arxiv.org/abs/2004.08997).
- [37] F. Sarghini, G. De Felice, and S. Santini, Neural networks based subgrid scale modeling in large eddy simulations, *Comput. Fluids* **32**, 97 (2003).
- [38] M. Gamahara and Y. Hattori, Searching for turbulence models by artificial neural network, *Phys. Rev. Fluids* **2**, 054604 (2017).
- [39] Z. Zhou, G. He, S. Wang, and G. Jin, Subgrid-scale model for large-eddy simulation of isotropic turbulent flows using an artificial neural network, *Comput. Fluids* **195**, 104319 (2019).
- [40] S. Pawar, O. San, A. Rasheed, and P. Vedula, A priori analysis on deep learning of subgrid-scale parameterizations for Kraichnan turbulence, *Theor. Comput. Fluid Dyn.* **34**, 429 (2020).
- [41] A. Volland, G. Balarac, and C. Corre, Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures, *J. Turbul.* **18**, 854 (2017).

- [42] C. 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).
- [43] C. 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).
- [44] C. J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, and T. Poinso, Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates, *Combust. Flame* **203**, 255 (2019).
- [45] S. Yellapantula, M. T. H. de Frahan, R. King, M. Day, and R. Grout, Machine Learning of Combustion LES Models from Reacting Direct Numerical Simulation, *Data Analysis for Direct Numerical Simulations of Turbulent Combustion* (Springer, New York, 2020), pp. 273–292.
- [46] S. Stolz and N. A. Adams, An approximate deconvolution procedure for large-eddy simulation, *Phys. Fluids* **11**, 1699 (1999).
- [47] R. Maulik and O. San, A neural network approach for the blind deconvolution of turbulent flows, *J. Fluid Mech.* **831**, 151 (2017).
- [48] Z. Yuan, C. Xie, and J. Wang, Modeling subgrid-scale stress by deconvolutional artificial neural networks in large eddy simulation of turbulence, *Phys. Fluids* **32**, 115106 (2020).
- [49] Y. Xie, E. Franz, M. Chu, and N. Thuerey, tempoGAN: A temporally coherent, volumetric gan for super-resolution fluid flow, *ACM Trans. Graphics (TOG)* **37**, 1 (2018).
- [50] K. Fukami, K. Fukagata, and K. Taira, Super-resolution reconstruction of turbulent flows with machine learning, *J. Fluid Mech.* **870**, 106 (2019).
- [51] K. Fukami, K. Fukagata, and K. Taira, Machine learning based spatio-temporal super resolution reconstruction of turbulent flows, *J. Fluid Mech.* **909**, A9 (2021).
- [52] A. Subramaniam, M. L. Wong, R. D. Borker, S. Nimmagadda, and S. K. Lele, Turbulence enrichment using generative adversarial networks, [arXiv:2003.01907](https://arxiv.org/abs/2003.01907).
- [53] H. Kim, J. Kim, S. Won, and C. Lee, Unsupervised deep learning for super-resolution reconstruction of turbulence, [arXiv:2007.15324](https://arxiv.org/abs/2007.15324).
- [54] M. Bode, M. Gauding, Z. Lian, D. Denker, M. Davidovic, K. Kleinheinz, J. Jitsev, and H. Pitsch, Using physics-informed super-resolution generative adversarial networks for subgrid modeling in turbulent reactive flows, [arXiv:1911.11380](https://arxiv.org/abs/1911.11380).
- [55] B. Siddani, S. Balachandar, W. Moore, Y. Yang, and R. Fang, Machine learning for physics-informed generation of dispersed multiphase flow using generative adversarial networks, [arXiv:2005.05363](https://arxiv.org/abs/2005.05363).
- [56] A. P. Singh, K. Duraisamy, and Z. J. Zhang, Augmentation of turbulence models using field inversion and machine learning, *55th AIAA Aerospace Sciences Meeting*, 2017, p. 0993.
- [57] S. Taghizadeh, F. Witherden, and S. Girimaji, Turbulence closure modeling with data-driven algorithms: Physical compatibility and consistency considerations, *New J. Phys.* **22**, 093023 (2020).
- [58] J. Sirignano, J. F. MacArt, and J. B. Freund, DPM: A deep learning PDE augmentation method with application to large-eddy simulation, *J. Comput. Phys.* **423**, 109811 (2020).
- [59] H. Raiesi, U. Piomelli, and A. Pollard, Evaluation of turbulence models using direct numerical and large-eddy simulation data, *J. Fluids Eng.* **133**, 021203 (2011).
- [60] R. L. Thompson, L. E. B. Sampaio, F. A. V. de Bragança Alves, L. Thais, and G. Mompean, A methodology to evaluate statistical errors in DNS data of plane channel flows, *Comput. Fluids* **130**, 1 (2016).
- [61] J. 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 (2019).
- [62] A. P. Singh and K. Duraisamy, Using field inversion to quantify functional errors in turbulence closures, *Phys. Fluids* **28**, 045110 (2016).
- [63] F. Köhler, J. Munz, and M. Schäfer, Data-driven augmentation of RANS turbulence models for improved prediction of separation in wall-bounded flows, *AIAA Scitech 2020 Forum*, 2020, p. 1586.
- [64] M. Yang and Z. Xiao, Improving the $k-\omega-A_r$ transition model by the field inversion and machine learning framework, *Phys. Fluids* **32**, 064101 (2020).
- [65] H. Xiao, J.-L. Wu, J.-X. Wang, R. Sun, and C. 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 (2016).

- [66] K. Duraisamy, Informing turbulence closures with computational and experimental data, *54th AIAA Aerospace Sciences Meeting*, 2016, p. 1556.
- [67] L. Franceschini, D. Sipp, and O. Marquet, Mean-flow data assimilation based on minimal correction of turbulence models: Application to turbulent high-Reynolds number backward-facing step, *Phys. Rev. Fluids* **5**, 094603 (2020).
- [68] L. Zhu, W. Zhang, J. Kou, and Y. Liu, Machine learning methods for turbulence modeling in subsonic flows around airfoils, *Phys. Fluids* **31**, 015105 (2019).
- [69] J. R. Holland, J. D. Baeder, and K. Duraisamy, Towards integrated field inversion and machine learning with embedded neural networks for RANS modeling, *AIAA Scitech 2019 Forum*, 2019, p. 1884.
- [70] J. R. Holland, J. D. Baeder, and K. Duraisamy, Field inversion and machine learning with embedded neural networks: Physics-consistent neural network training, *AIAA Aviation 2019 Forum*, 2019, p. 3200.
- [71] M. B. Giles, M. C. Duta, J.-D. Müller, and N. A. Pierce, Algorithm developments for discrete adjoint methods, *AIAA J.* **41**, 198 (2003).
- [72] P. J. Blonigan and Q. Wang, Multiple shooting shadowing for sensitivity analysis of chaotic dynamical systems, *J. Comput. Phys.* **354**, 447 (2018).
- [73] A. Ni, Q. Wang, P. Fernandez, and C. Talnikar, Sensitivity analysis on chaotic dynamical systems by finite difference non-intrusive least squares shadowing (FD-NILSS), *J. Comput. Phys.* **394**, 615 (2019).
- [74] W. Liu and J. Fang, Iterative framework of machine-learning based turbulence modeling for reynolds-averaged navier-stokes simulations, [arXiv:1910.01232](https://arxiv.org/abs/1910.01232).
- [75] Y. 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).
- [76] B. D. Tracey, K. Duraisamy, and J. J. Alonso, A machine learning strategy to assist turbulence model development, *53rd AIAA Aerospace Sciences Meeting*, Vol. 2015-1287, 2015.
- [77] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, *J. Mach. Learn. Res.* **3**, 1157 (2003).
- [78] J. A. Langford and R. D. Moser, Optimal LES formulations for isotropic turbulence, *J. Fluid Mech.* **398**, 321 (1999).
- [79] E. J. Parish and K. Duraisamy, Non-Markovian closure models for large eddy simulations using the Mori-Zwanzig formalism, *Phys. Rev. Fluids* **2**, 014604 (2017).
- [80] T. Lund, The use of explicit filters in large eddy simulation, *Comput. Math. Appl.* **46**, 603 (2003).
- [81] J. Mathew, H. Foysi, and R. Friedrich, A new approach to LES based on explicit filtering, *Int. J. Heat Fluid Flow* **27**, 594 (2006).
- [82] M. Du, N. Liu, and X. Hu, Techniques for interpretable machine learning, *Commun. ACM* **63**, 68 (2019).
- [83] L. Jofre, S. P. Domino, and G. Iaccarino, A framework for characterizing structural uncertainty in large-eddy simulation closures, *Flow, Turbul. Combust.* **100**, 341 (2018).