

Enhancing Computational Fluid Dynamics with Machine Learning

Ricardo Vinuesa^{1,2*} and Steven L. Brunton³

¹ FLOW, Engineering Mechanics, KTH Royal Institute of Technology, Stockholm, Sweden

² Swedish e-Science Research Centre (SeRC), Stockholm, Sweden

³ Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, United States

* Corresponding author: Ricardo Vinuesa (rvinuesa@mech.kth.se)

Abstract

Machine learning is rapidly becoming a core technology for scientific computing, with numerous opportunities to advance the field of computational fluid dynamics. In this Perspective, we highlight some of the areas of highest potential impact, including to accelerate direct numerical simulations, to improve turbulence closure modeling, and to develop enhanced reduced-order models. We also discuss emerging areas of machine learning that are promising for computational fluid dynamics, as well as some potential limitations that should be taken into account.

1 Introduction

The field of numerical simulation of fluid flows is generally known as computational fluid dynamics (CFD). Fluid mechanics is an area of great importance, both from a scientific perspective and for a range of industrial-engineering applications. Fluid flows are governed by the Navier–Stokes equations, which are partial differential equations (PDEs) modeling the conservation of mass and momentum in a Newtonian fluid. These PDEs are non-linear due to the convective acceleration (which is related to the change of velocity with the position), and they commonly exhibit time-dependent chaotic behavior, known as *turbulence*. Solving the Navier–Stokes equations for turbulent flows requires numerical methods that may be computationally expensive, or even intractable at high Reynolds numbers, due to the wide range of scales in space and time necessary to resolve these flows. There are various approaches to numerically solve these equations, which can be discretized using methods of different orders, for instance finite-difference [1], finite-volume [2], finite-element [3], spectral methods [4], and so forth. Furthermore, turbulence can be simulated with different levels of fidelity and computational cost.

At the same time, we are experiencing a revolution in the field of machine learning (ML), which is enabling advances across a wide range of scientific and engineering areas [5–9]. Machine learning is a subfield of the broader area of artificial intelligence (AI), which is focused on the development of algorithms with the capability of learning from data without explicit mathematical models [10]. Many of the most exciting advances in ML have leveraged deep learning, based on neural networks (NNs) with multiple hidden layers between the input and the output. One key aspect contributing to the remarkable success of deep learning is the ability to learn in a hierarchical manner: while initial layers learn simple relationships in the data, deeper layers combine this information to learn more abstract relationships. Many physical problems exhibit this hierarchical behavior, and can therefore be effectively modelled using deep learning, and machine learning more generally.

In this Perspective, we focus on the potential of machine learning to improve CFD, including possibilities to increase the speed of high-fidelity simulations, develop turbulence models with different levels of fidelity, and produce reduced-order models beyond what can be achieved with classical approaches. Several authors have surveyed the potential of machine learning to improve fluid mechanics [11, 12], including topics beyond the scope of CFD, such as experimental techniques, control applications, and related fields. Others have reviewed more specific aspects of ML for CFD, such as turbulence closure [13, 14] and heat-transfer aspects of CFD for aerodynamic optimization [15]. Our discussion will address the middle ground of ML for CFD more broadly, with a schematic representation of topics covered in Fig. 1. Approaches to improve CFD with ML are aligned with the larger efforts to incorporate ML into scientific computing, for instance via physics-informed neural networks (PINNs) [16, 17] or to accelerate computational chemistry [8, 18].

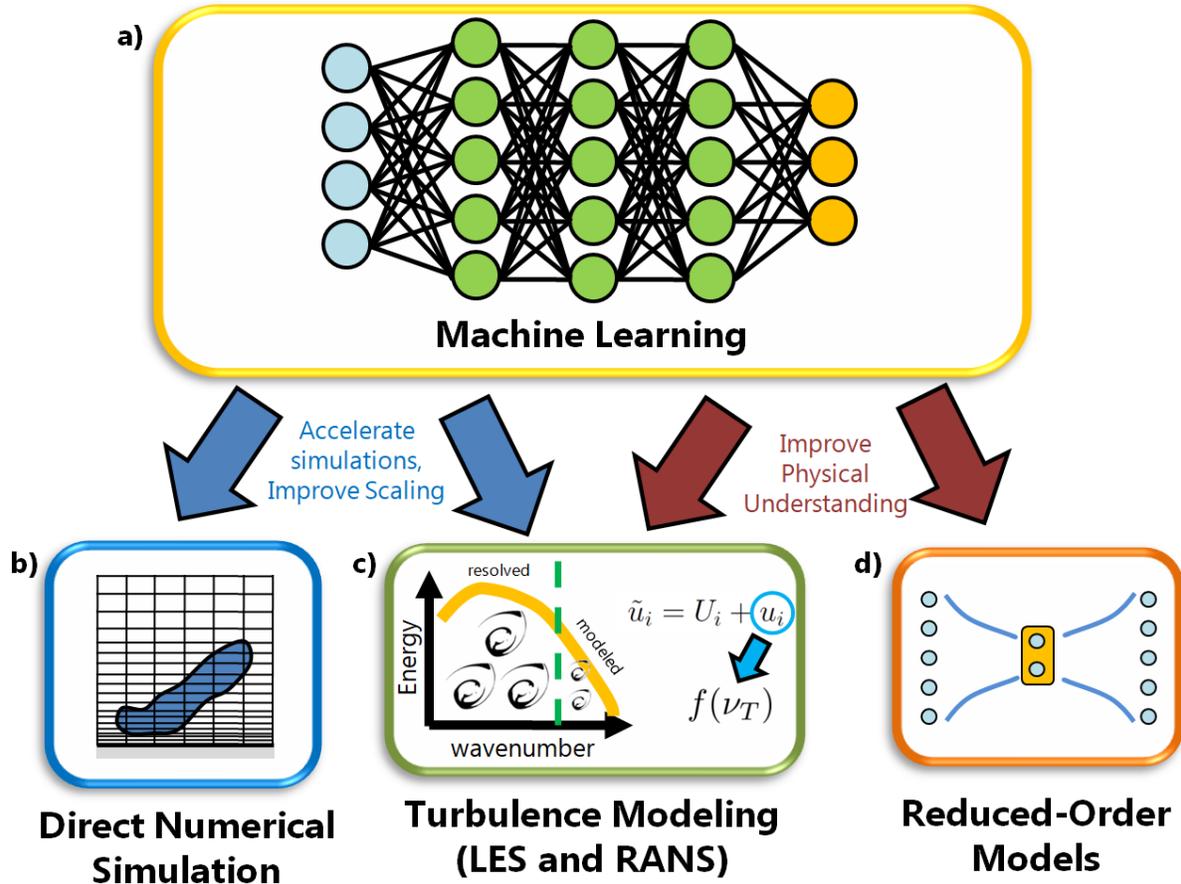


Figure 1: **Summary of some of the most relevant areas where machine learning can enhance CFD.** a) Neural network illustrating the field of machine learning. b) In direct numerical simulations, computational grids fine enough to properly resolve the details of the flow structures (such as the one shown in blue) are needed. The grey band denotes the wall. c) The left subpanel illustrates large-eddy simulation (LES), and we show a schematic of the turbulence spectrum, where the larger scales (to the left of the green vertical line, *i.e.* the cut-off wavenumber) are numerically simulated, and the smaller ones (to the right of the cut-off) are represented by a model. Since these smaller scales do not need to be resolved, the computational cost of the simulation is reduced. The right subpanel illustrates Reynolds-averaged Navier–Stokes (RANS) simulations, where \tilde{u}_i is the instantaneous velocity component i , U_i the mean and u_i the fluctuating component. In RANS models, the fluctuations are usually expressed as a function of the eddy viscosity ν_T . d) General schematic of a dimensionality-reduction method, where the circles denote flow quantities. The input (on the left in blue) is the high-resolution flow field, the output (on the right in red) is the reconstruction of the input field, and the yellow box represents the low-dimensional system.

2 Accelerating direct numerical simulations

Direct numerical simulation (DNS) is a high-fidelity approach where the governing Navier–Stokes equations are discretized and integrated in time with enough degrees of freedom to resolve all flow structures. Turbulent flows exhibit a pronounced multi-scale character, with vortical structures across a range of sizes and energetic content [19]. This complexity requires fine meshes and accurate computational methods to avoid distorting the underlying physics with numerical artifacts. With a properly designed DNS, it is possible to obtain a representation of the flow field with the highest level of detail among CFD methods. However, the fine computational meshes required to resolve the smallest scales lead to exceedingly high computational costs, which increase with the Reynolds number [20].

A number of machine-learning approaches have been developed recently to improve the efficiency of DNS. First, we will discuss several studies aimed at improving discretization schemes. Bar-Sinai *et al.* [21] proposed a technique based on deep learning to estimate spatial derivatives in low-resolution grids, outperforming standard finite-difference methods. A similar approach was developed by Stevens and Colonius [22] to improve the

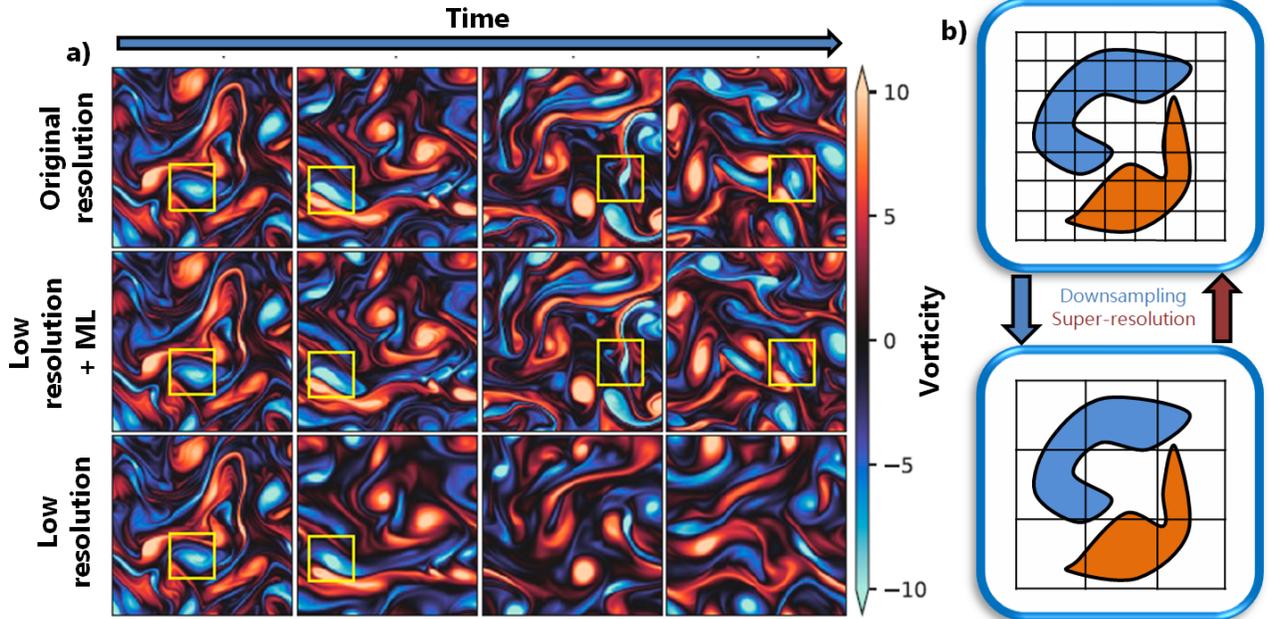


Figure 2: **An example of machine-learning-accelerated direct numerical simulation.** a) Results from the work by Kochkov *et al.* [25], where the instantaneous vorticity field is shown for simulations with high-/low-resolution, as well as a case with low resolution supplemented with ML. Four different time steps are shown, and some key vortical structures are highlighted with yellow squares. Adapted from the work by Kochkov *et al.* [25], with permission of the publisher (United States National Academy of Sciences). b) Sketch represents that ML accelerates the simulation through a correction between the coarse and the fine resolutions. In this example, one can reduce the resolution by downsampling the flow data on a coarser mesh, and then use ML to recover the details present in the finer simulation through super-resolution. The correction between coarse and fine resolutions, which is based on ML, enables the super-resolution task here.

results of fifth-order finite-difference schemes in the context of shock-capturing simulations. Jeon and Kim [23] proposed to use a deep neural network to simulate the well-known finite-volume discretization scheme [2] employed in fluid simulations. They tested their method with reactive flows, obtaining very good agreement with reference high-resolution data at one tenth the computational cost. However, they also documented errors with respect to the reference solution which increased with time. Another deep-learning approach, based on a fully-convolutional/long-short-term-memory (LSTM) network, was proposed by Stevens and Colonius [24] to improve the accuracy of finite-difference/finite-volume methods. Second, we consider the strategy of developing a correction between fine- and coarse-resolution simulations. This was developed by Kochkov *et al.* [25] for the two-dimensional Kolmogorov flow [26], which maintains fluctuations via a forcing term. They leveraged deep learning to develop the correction, obtaining excellent agreement with reference simulations in meshes from 8 to 10 times coarser in each dimension, as shown in Fig. 2. In particular, this figure shows that for long simulations, the standard coarse-resolution case does not exhibit certain important vortical motions, which are however properly captured by the low-resolution case with the ML model. These results promise to substantially reduce the computational cost of relevant fluid simulations, including weather [27], climate [28], engineering [29] and astrophysics [30]. Finally, other strategies to improve the performance of PDE solvers in coarser meshes have been developed by Li *et al.* [31–33].

It is also possible to accelerate CFD by solving the Poisson equation with deep learning; this has been done by several research groups in various areas, such as simulations of electric fields [34] and particles [35]. The Poisson equation is frequently used in operator-splitting methods to discretize the Navier–Stokes equations [36], where first the velocity field is advected, and the resulting field \mathbf{u}^* does not satisfy the continuity equation (*i.e.* for incompressible flows, \mathbf{u}^* does not satisfy the divergence-free condition). The second step involves a correction to ensure that \mathbf{u}^* is divergence free, leading to the following Poisson equation:

$$\frac{\Delta t}{\rho} \nabla^2 p = -\nabla \cdot \mathbf{u}^*, \quad (1)$$

where Δt is the simulation time step, ρ is the fluid density, and p is the pressure. Solving this equation is typically the most computationally expensive step of the numerical solver. Therefore, devising alternative strategies to solve

it more efficiently is an area of great promise. Machine learning can be used for this task, because this technology can exploit data from previous examples to find mappings between the divergence of the uncorrected velocity and the resulting pressure field. For instance, Ajuria *et al.* [37] proposed using a convolutional neural network (CNN) to solve Equation (1) in incompressible simulations, and tested it in a plume configuration. Their results indicate that it is possible to outperform the traditional Jacobi solver with good accuracy at low Richardson numbers Ri (the Richardson number measures the ratio between the buoyancy and the shear in the flow). However, the accuracy degrades at higher Ri , motivating the authors to combine the CNN with the Jacobi solver (which is used when the CNN diverges). CNNs were also used to solve the Poisson problem by decomposing the original problem into a homogeneous Poisson problem plus four inhomogeneous Laplace subproblems [38]. This decomposition resulted in errors below 10%, which motivates using this approach as a first guess in iterative algorithms, potentially reducing computational cost. Other data-driven methods [39] have been proposed to accelerate the pressure correction in multi-grid solvers. These approaches may also be used to accelerate simulations of lower fidelity that rely on turbulence models.

Numerical simulations can also be accelerated by decreasing the size of the computational domain needed to retain physical properties of the system. Two ways in which the domain can be reduced are to replace a section upstream of the domain of interest by an inflow condition, and to replace part of the far-field region by a suitable boundary condition. Doing so, these parts of the domain do not need to be simulated, thus producing computational savings, and ML can help to develop the inflow and far-field conditions as discussed next. Fukami *et al.* [40] developed a time-dependent inflow generator for wall-bounded turbulence simulations using a convolutional autoencoder with a multilayer perceptron (MLP). They tested their method in a turbulent channel flow at $Re_\tau = 180$, which is the friction Reynolds number based on channel half height and friction velocity, and they maintained turbulence for an interval long enough to obtain converged turbulence statistics. This is a promising research direction due to the fact that current inflow-generation methods show limitations in terms of generality of the inflow conditions, for instance at various flow geometries and Reynolds numbers. A second approach to reduce the computational domain in external flows is to devise a strategy to set the right pressure-gradient distribution without having to simulate the far field. This was addressed by Morita *et al.* [41] through Bayesian optimization based on Gaussian-process regression, achieving accurate results when imposing concrete pressure-gradient conditions on a turbulent boundary layer.

3 Improving turbulence models

DNS is impractical for many real-world applications due to the computational cost associated with resolving all scales for flows with high Reynolds numbers, together with difficulties arising from complex geometries. Industrial CFD typically relies on either Reynolds-averaged Navier–Stokes (RANS) models, where no turbulent scales are simulated, or coarsely-resolved large-eddy simulations (LES), where only the largest turbulent scales are resolved and smaller ones are modelled. Here the term *model* refers to an *a-priori* assumption regarding the physics of a certain range of turbulent scales. In the following we discuss ML applications to RANS and LES modeling.

3.1 RANS modeling

Numerical simulations based on RANS models rely on the so-called RANS equations, which are obtained after decomposing the instantaneous flow quantities into a mean and a fluctuating component, and averaging the Navier–Stokes equations in time. Using index notation, the instantaneous i -th velocity component \tilde{u}_i can be decomposed into its mean (U_i) and fluctuating (u_i) components as follows: $\tilde{u}_i = U_i + u_i$. Although the RANS equations govern the mean flow, the velocity fluctuations are also present in the form of the Reynolds stresses $\overline{u_i u_j}$ (where the overbar denotes averaging in time), which are correlations between the i -th and j -th velocity components. Since it is convenient to derive equations containing only mean quantities, $\overline{u_i u_j}$ needs to be expressed as a function of the mean flow, and this is called the *closure problem*. The first approach to do this was proposed by Boussinesq [42], who related the Reynolds stresses with the mean flow via the so-called eddy viscosity ν_T . Although this approach has led to some success, there are still a number of open challenges for RANS modeling in complex flows [43], where this approach is too simple. In particular, conventional RANS models exhibit significant errors when dealing with complex pressure-gradient distributions, complicated geometries involving curvature, separated flows, flows with significant degree of anisotropy and three-dimensional effects, among others. As argued by Kutz [44], machine learning can produce more sophisticated models for the Reynolds stresses by using adequate data, in particular if the examples used for training represent a sufficiently rich set of flow configurations.

A wide range of ML methods have recently been used to improve RANS turbulence modeling [13], focusing

on the challenge of improving the accuracy of the Reynolds stresses for general conditions. For example, Ling *et al.* [45] proposed a novel architecture, including a multiplicative layer with an invariant tensor basis, used to embed Galilean invariance in the predicted anisotropy tensor. Incorporating this invariance improves the performance of the network, which outperforms traditional RANS models based on linear [42] and nonlinear [46] eddy-viscosity models. They tested their models for turbulent duct flow and the flow over a wavy wall, which are challenging to predict with RANS models [47, 48] because of the presence of secondary flows [49]. Other ML-based approaches [50, 51] rely on physics-informed random forests to improve RANS models, with applications to cases with secondary flows and separation. On the other hand, Jiang *et al.* [52] recently developed an interpretable framework for RANS modeling based on a physics-informed residual network (PiResNet). Their approach relies on two modules to infer the structural and parametric representations of turbulence physics, and includes non-unique mappings, a realizability limiter, and noise-insensitivity constraints. Interpretable models are essential for engineering and physics, particularly in the context of turbulence modeling. The interpretability of the framework by Jiang *et al.* [52] relies on its constrained model form [53], although this is not generally possible to attain. Recent studies [54] have discussed various approaches to include interpretability in the development of deep-learning models, and one promising approach is the one proposed by Cranmer *et al.* [55]. This technique has potential in terms of embedding physical laws and improving our understanding of such phenomena. Other interpretable RANS models were proposed by Weatheritt and Sandberg [56], using gene-expression programming (GEP), which is a branch of evolutionary computing [57]. GEP iteratively improves a population of candidate solutions by survival of the fittest, with the advantage of producing closed-form models. The Reynolds-stress anisotropy tensor was also modelled by Weatheritt and Sandberg [58], who performed tests in RANS simulations of turbulent ducts. Models based on sparse identification of nonlinear dynamics (SINDy) [59] have also been used for RANS closure models [60–62].

Furthermore, the literature also reflects the importance of imposing physical constraints in the models and incorporating uncertainty-quantification (UQ) [63–65] alongside ML-based models. It is also important to note that when using DNS quantities to replace terms in the RANS closure, the predictions may be unsatisfactory [66]. This inadequacy is due to the strict assumptions associated with the RANS model, as well as the potential ill-conditioning of the RANS equations [67]. It is thus essential to take advantage of novel data-driven methods, while also ensuring that uncertainties are identified and quantified. Another interesting review by Ahmed *et al.* [14] discussed both classical and emerging data-driven closure approaches, also connecting with ROMs.

Obiols-Sales *et al.* [68] developed a method to accelerate the convergence of RANS simulations based on the very popular Spalart–Allmaras (SA) turbulence model [69] using the CFD code OpenFOAM [70]. In essence, they combined iterations from the CFD solver and evaluation of a CNN model, obtaining convergence from 1.9 to 7.4 times faster than that of the CFD solver, both in laminar and turbulent flows. Multiphase flows, which consist of flows with two or more thermodynamic phases, are also industrially relevant. Gibou *et al.* [71] proposed different directions in which machine learning and deep learning can improve CFD of multiphase flows, in particular when it comes to enhancing the simulation speed. Ma *et al.* [72] used deep learning to predict the closure terms (i.e., gas flux and streaming stresses) in their two-fluid bubble flow, whereas Mi *et al.* [73] analyzed gas-liquid flows and employed neural networks to identify the different flow regimes.

3.2 LES modeling

LES-based numerical simulations rely on low-pass filtering the Navier–Stokes equations (Fig. 1c), such that the largest scales are simulated and the smallest ones (below a certain cut-off) are modeled by means of a so-called subgrid-scale model (SGS). Note that the smallest scales are the most demanding from a computational point of view (both in terms of computer time and memory usage), because they require fine meshes to be properly resolved. The first proponent of this type of approach was Smagorinski [74], who developed an SGS model based on an eddy viscosity which was computed in terms of the mean flow and the grid size; his model assumed that the production equals the dissipation for the small scales. Although LES can lead to substantial computational savings with respect to DNS while exhibiting a good accuracy, there are still challenges associated with its usage for general purpose [43]. For instance, current SGS models exhibit limited accuracy in their predictions of turbulent flows at high Reynolds number, in complex geometries, and in cases with shocks and chemical reactions.

Machine learning has also been used to develop SGS models in the context of LES of turbulent flows basically in two ways: supplementing the unresolved energy in the coarse mesh using supervised learning and developing agent-based approaches to stabilize the coarse simulation. When it comes to the first approach, we list several studies below which rely on high-fidelity data to train both deep-learning and GEP-based models. First, Beck *et al.* [75] used an artificial neural network based on local convolutional filters to predict the mapping between the flow in a coarse simulation and the closure terms, using a filtered DNS of decaying homogeneous isotropic turbulence. Lapeyre *et al.* [76] employed a similar approach, with a CNN architecture inspired by a U-net model, to predict the

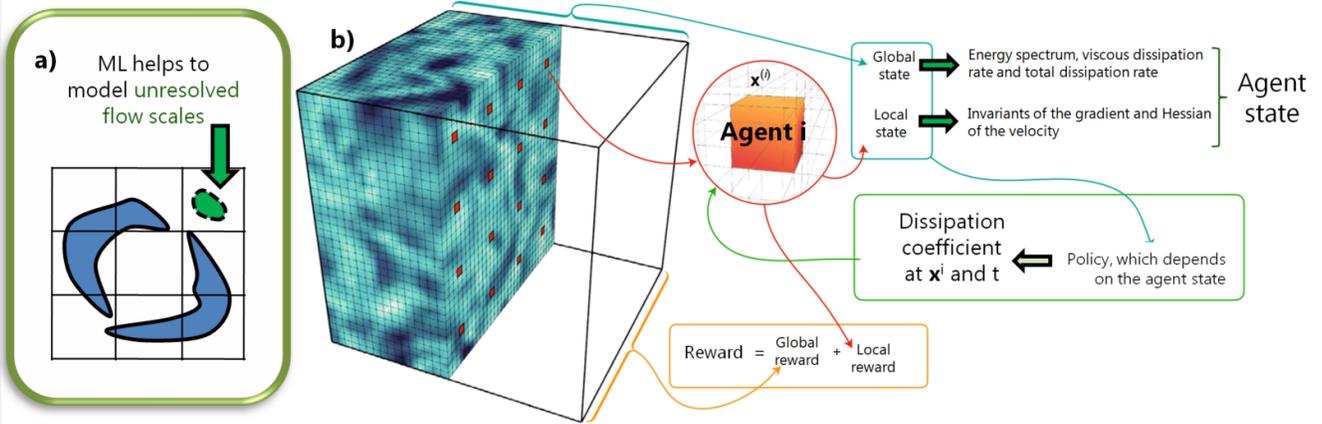


Figure 3: **An example of LES modeling where the dissipation coefficient in the Smagorinski model is calculated by means of machine learning.** a) Schematic of a numerical simulation fine enough to resolve the blue flow structures, but too coarse to resolve the green structure. This small-scale structure (green) would have to be modeled by means of a subgrid-scale (SGS) model, such as the well-known SGS model by Smagorinski [74]. This model relies on an empirical parameter, the dissipation coefficient C_s , which can be determined more accurately by ML. b) An example of ML-based approach to determine the value of C_s by reinforcement learning (RL) [83]. The RL agents are located at the red blocks, and they are used to compute C_s for each grid point with coordinates x^i at time t . This calculation is carried out in terms of a policy, which depends on the state of the agent. This state is based on local variables (invariants of the gradient and Hessian of the velocity) and also global ones (energy spectrum, viscous dissipation rate and total dissipation rate). The agents receive a reward, which also depends on local and global quantities, based on the accuracy of the simulation. Adapted from the work by Novati *et al.* [83], with permission of the publisher (Springer Nature).

subgrid-scale wrinkling of the flame surface in premixed turbulent combustion; they obtained better results than classical algebraic models. Maulik *et al.* [77] employed a multilayer perceptron (MLP) to predict the SGS model in an LES using high-fidelity numerical data to train the model. They evaluated the performance of their method on Kraichnan turbulence [78], which is a classical two-dimensional decaying-turbulence test case. Several other studies have also used neural networks in a supervised manner for SGS modeling [79–81]. Furthermore, GEP has also been used for SGS modeling [82] in an LES of a Taylor–Green vortex, outperforming standard LES models. Regarding the second approach to LES modeling, Novati *et al.* [83] employed multi-agent reinforcement-learning (RL) to estimate the unresolved subgrid-scale physics. This unsupervised method exhibits favorable generalization properties across grid sizes and flow conditions, and the results are presented for isotropic turbulence. As shown in the schematic representation of Fig. 3, the state of the agents at a particular instant is given in terms of both local and global variables; this state is then used to calculate the so-called dissipation coefficient at each grid point.

In certain applications, for instance those involving atmospheric boundary layers (ABLs), the Reynolds number is several orders of magnitude larger than those of most studies based on turbulence models or wind-tunnel experiments [84]. The mean flow in the inertial sublayer has been widely studied in the ABL community, and it is known that in neutral conditions it can be described by a logarithmic law [85]. The logarithmic description of the inertial sublayer led to the use of wall models, which replace the region very close to the wall with a model defining a surface shear stress matching the logarithmic behavior. This is the cornerstone of most atmospheric models, which avoid resolving the computationally expensive scales close to the wall. One example is the work by Giometto *et al.* [86], who studied a real urban geometry, adopting the LES model by Bou-Zeid *et al.* [87] and the Moeng model [88] for the wall boundary condition. It is possible to use data-driven approaches to develop mappings between the information in the outer region (which is resolved) and a suitable off-wall boundary condition (so the near-wall region does not need to be resolved). For instance, it is possible to exploit properties of the logarithmic layer and rescale the flow in the outer region to set the off-wall boundary condition in turbulent channels [89, 90]. This may also be accomplished via transfer functions in spectral space [91], convolutional neural networks [92], or modeling the temporal dynamics of the near-wall region via deep neural networks [93]. Other promising approaches based on deep learning are the one by Moriya *et al.* [94], based on defining a virtual velocity, and the reinforcement-learning technique by Bae and Koumoutsakos [95]. Defining off-wall boundary conditions with machine learning is a challenging yet promising area of research.

4 Developing reduced-order models

Machine learning is also being used to develop reduced-order models (ROMs) in fluid dynamics. ROMs rely on the fact that even complex flows often exhibit a few dominant coherent structures [96–98] that may provide coarse, but valuable information about the flow. Thus, ROMs describe the evolution of these coherent structures, providing a lower-dimensional, lower-fidelity characterization of the fluid. In this way, ROMs provide a fast surrogate model for the more expensive CFD techniques described above, enabling optimization and control tasks that rely on many model iterations or fast model predictions. The cost of this efficiency is a loss of generality: ROMs are tailored to a specific flow configuration, providing massive acceleration but a limited range of applicability.

Developing a reduced-order model involves (1) finding a set of reduced *coordinates*, typically describing the amplitudes of important flow structures, and (2) identifying a differential-equation model (*i.e.*, a dynamical system) for how these amplitudes evolve in time. Both of these stages have seen incredible recent advances with machine learning. One common ROM technique involves learning a low-dimensional coordinate system with the proper orthogonal decomposition (POD) [96, 99] and then obtaining a dynamical system for the flow system restricted to this subspace by Galerkin projection of the Navier–Stokes equations onto these modes. Although the POD step is data driven, working equally well for experiments and simulations, Galerkin projection requires a working numerical implementation of the governing equations; moreover, it is often *intrusive*, involving custom modifications to the numerical solver. The related dynamic-mode decomposition (DMD) [100] is a purely data-driven procedure that identifies a low-dimensional subspace and a linear model for how the flow evolves in this subspace. Here, we will review a number of recent developments to extend these approaches with machine learning.

The first broad opportunity to incorporate machine learning into ROMs is in learning an improved coordinate system in which to represent the reduced dynamics. POD [96, 99] provides an orthogonal set of modes that may be thought of as a data-driven generalization of Fourier modes, which are tailored to a specific problem. POD is closely related to principal-component analysis (PCA) and the singular-value decomposition (SVD) [5], which are two core dimensionality-reduction techniques used in data-driven modeling. These approaches provide linear subspaces to approximate data, even though it is known that many systems evolve on a nonlinear manifold. Deep learning provides a powerful approach to generalize the POD/PCA/SVD dimensionality reduction from learning a linear subspace to learning coordinates on a curved manifold. Specifically, these coordinates may be learned using a neural network *autoencoder*, which has an input and output the size of the high-dimensional fluid state and a constriction or bottleneck in the middle that reduces to a low-dimensional latent variable. The map from the high-dimensional state \mathbf{x} to the latent state \mathbf{z} is called the *encoder* and the map back from the latent state to an estimate of the high-dimensional state $\hat{\mathbf{x}}$ is the *decoder*. The autoencoder loss function is $\|\hat{\mathbf{x}} - \mathbf{x}\|_2^2$. When the encoder and decoder consist of a single layer and all nodes have identity activation functions, then the optimal solution to this network will be closely related to POD [101]. However, this shallow linear autoencoder may be generalized to a deep nonlinear autoencoder with multiple encoding and decoding layers and nonlinear activation functions for the nodes. In this way, a deep autoencoder learns nonlinear manifold coordinates that may considerably improve the compression in the latent space, with increasing applications in fluid mechanics [102, 103]. This concept is illustrated in Fig. 4 for the simple flow past a cylinder, where it is known that the energetic coherent structures evolve on a parabolic sub-manifold in the POD subspace [104]. Lee and Carlberg [105] recently showed that deep convolutional autoencoders may be used to greatly improve the performance of classical ROM techniques based on linear subspaces [106, 107]. More recently, Cenedese *et al.* [108] proposed a promising data-driven method to construct ROMs on spectral submanifolds (SSMs).

Once an appropriate coordinate system is established, there are many machine-learning approaches to model the dynamics in these coordinates. Several neural networks are capable of learning nonlinear dynamics [109], including the LSTM network [110–112] and echo-state networks [113], which are a form of reservoir computing. Beyond neural networks, there are alternative regression techniques to learn effective dynamical-systems models. Cluster reduced-order modeling (CROM) [114] is a simple and powerful unsupervised-learning approach that decomposes a time series into a few representative clusters, and then models the transition-state probability between clusters. The operator-inference approach is closely related to Galerkin projection, where a neighboring operator is learned from data [115–117]. The sparse identification of nonlinear dynamics (SINDy) [59] procedure learns a minimalistic model by fitting the observed dynamics to the fewest terms in a library of candidate functions that might describe the dynamics, resulting in models that are interpretable and balance accuracy and efficiency. SINDy has been used to model a range of fluids [118–124], including laminar and turbulent wake flows, convective flows, and shear flows.

The modeling approaches above may be combined with a deep autoencoder to uncover a low-dimensional latent space, as in the SINDy-autoencoder [125]. DMD has also been extended to nonlinear coordinate embeddings through the use of deep-autoencoder networks [126–130]. In all of these architectures, there is a tremendous

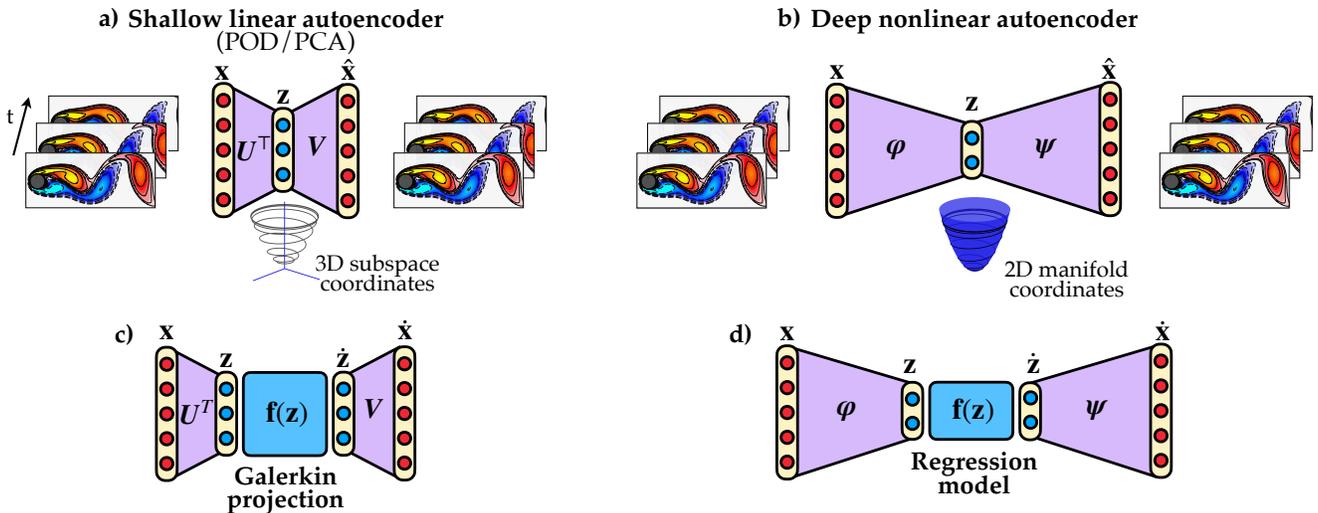


Figure 4: **Schematic of neural-network autoencoders for dimensionality reduction and model identification.** Reduced-order models are typically designed to balance efficiency and accuracy. ML solutions further improve the efficiency by reducing the effective dimension of the model and increasing the accuracy through better modeling of how these few variables co-evolve in time. In this figure, the input is a high-resolution flow field evolving in time (t) and the output is a reconstruction of that field from the latent space. a) Classic proper orthogonal decomposition/principal component analysis (POD/PCA) may be viewed as a shallow autoencoder with a single encoder U^T and decoder V layers, together with linear activation units. For the flow past a cylinder example shown, the dynamics evolve in a three-dimensional coordinate system. b) A deep, multi-level autoencoder with multi-layer encoder φ and decoder ψ , as well as nonlinear activation functions provides enhanced nonlinear coordinates on a manifold. The cylinder flow now evolves on a two-dimensional submanifold. c) The classic Galerkin-projection model, obtained by projecting the governing Navier–Stokes equations onto an orthogonal basis. The Galerkin-projection model in c) can be replaced by more generic machine-learning regressions in d), such as long-short-term-memory (LSTM) networks, reservoir networks or sparse nonlinear models to represent the nonlinear dynamical system $\dot{z} = f(z)$.

opportunity to embed partial knowledge of the physics, such as conservation laws [118], symmetries [120], and invariances [131–133]. It may also be possible to directly impose stability in the learning pipeline [118, 134, 135].

The ultimate goal for machine-learning ROMs is to develop models that have improved accuracy and efficiency, better generalizability to new initial and boundary conditions, flow configurations, varying parameters, as well as improved model interpretability, ideally with less intrusive methods and less data. Enforcing partially-known physics, such as symmetries and other invariances, along with sparsity, is expected to be critical in these efforts. It is also important to continue integrating these efforts with the downstream applications of control and optimization. Finally, many applications of fluid dynamics involve safety-critical systems, and therefore certifiable models are essential.

5 Emerging possibilities and outlook

In this Perspective we have provided our view on the potential of ML to advance the capabilities of CFD, focusing on three main areas: accelerating simulations, enhancing turbulence models, and improving reduced-order models. In Table 1 we highlight some examples of application within each of the three areas. There are also several emerging areas of ML that are promising for CFD, which we discuss in this section. One area is non-intrusive sensing - that is the possibility of performing flow predictions based on, for instance, information at the wall. This task, which has important implications for closed-loop flow control [136], has been carried out via CNNs in turbulent channels [137]. In connection to the work by Guastoni *et al.* [137], there are a number of studies documenting the possibility of performing super-resolution predictions (*e.g.* when limited flow information is available) in wall-bounded turbulence using CNNs, autoencoders and generative-adversarial networks (GANs) [138–141]. Another promising direction is the imposition of constraints based on physical invariances and symmetries on the ML model, which has been used for SGS modeling [131], ROMs [118], and for geophysical flows [133].

Accelerate DNS	Improve LES/RANS	Develop ROMs	Future developments
Deep learning for correction between coarse and fine simulations	Deep learning / reinforcement learning for more general subgrid-scale models	Proper-orthogonal decomposition for improved linear subspaces to represent dynamics	Deep learning for prediction and control of turbulent flows
Deep learning for accelerating the solution of the Poisson equation	Deep learning / reinforcement learning for more robust wall wall models	Autoencoders to learn manifolds to represent dynamics	Better interpretability of deep learning for computational-fluid-dynamics models
Domain reduction: deep learning for inflow generation; Gaussian processes for pressure-gradient distributions	Deep learning / symbolic regression for more accurate and general RANS models	Deep learning for representing temporal system dynamics	More efficient operation of high-performance-computing facilities facilitated by ML
Deep learning for improved accuracy of finite-difference / finite-volume schemes	Deep learning / symbolic regression for interpretable RANS models	Sparse identification of nonlinear dynamics for learning efficient and interpretable ROMs	ML-enabled more advanced computational architectures

Table 1: **Summary of applications of ML to enhance CFD.** We show options to accelerate direct numerical simulations (DNSs), improvement of turbulence models for large-eddy simulations (LESs) and Reynolds-averaged Navier–Stokes (RANS) simulations, development of reduced-order models (ROMs) and possible future developments.

Physics-informed neural networks (PINNs) constitute another family of methods that are becoming widely adopted for scientific computing, more broadly. This technique, introduced by Raissi *et al.* [16], uses deep learning to solve PDEs, exploiting the concept of automatic differentiation used in the back-propagation algorithm to calculate partial derivatives and form the equations, which are enforced through a loss function. Although the method is different from the ML approaches discussed above, in certain cases PINNs provide a promising alternative to traditional numerical methods for solving PDEs. This framework also shows promise for biomedical simulations, in particular after the recent work by Raissi *et al.* [142], in which the concentration field of a tracer is used as an input to accurately predict the instantaneous velocity fields by minimizing the residual of the Navier–Stokes equations. The growing usage of PINNs highlights the relevance of exploiting the knowledge we have about the governing equations when conducting flow simulations. In fact, PINNs are being used for turbulence modeling [143] (solving the RANS equations without the need of a model for the Reynolds stresses), for developing ROMs [144], for dealing with noisy data [145] and for accelerating traditional solvers, for instance by means of solving the Poisson equation more efficiently [146].

There are also grand challenges in CFD that necessitate new methods in ML. One motivating challenge in CFD is to perform accurate coarse-resolution simulations in unforced three-dimensional wall-bounded turbulent flows. The production of turbulent kinetic energy (TKE) in these flows takes place close to the wall [147] (although at very high Reynolds numbers, outer-layer production also becomes relevant), and therefore using coarse meshes may significantly distort TKE production. In these flows of high technological importance, the TKE production sustains the turbulent fluctuations, and therefore a correction between coarse and fine resolutions may not be sufficient to obtain accurate results. Another challenge is the need for robust and interpretable models, a goal which is not easily attainable with deep-learning methods. There are however promising directions [54, 55] to achieve interpretable deep-learning models, with important applications to CFD. A clear challenge of CFD is the large energy consumption associated with large-scale simulations. In this sense, ML can be an enabler for more efficient operation of the high-performance-computing (HPC) facilities [7], or even for future computational architectures, including quantum computers [148], as shown in Table 1.

It is worth noticing that, while ML has a very high potential for CFD, there are also a number of caveats which may limit the applicability of ML to certain areas of CFD. Firstly, ML methods, such as deep learning, are often expensive to train and require large amounts of data. It is therefore important to identify areas where ML outperforms classical methods, which have been established for decades, and may be more accurate and efficient in certain cases. For instance, it is possible to develop interpretable ROMs with traditional methods such as POD and DMD, and while deep learning can provide some advantages [103], the simpler machine-learning methods may be more efficient and straightforward. Finally, it is important to assess the information about the training data available to

the user: certain flow properties (e.g. incompressibility, periodicity, etc.) should be embedded in the ML model to increase training efficiency and prediction accuracy. There is also a question of how the training data is generated, and whether the associated cost is taken into account when benchmarking. In this context, transfer learning is a promising area [137].

Despite the caveats above, we believe that the trend of advancing CFD with ML will continue in the future. This progress will continue to be driven by an increasing availability of high-quality data, high-performance computing, and a better understanding and facility with these emerging techniques. Improved adoption of reproducible research standards [149, 150] is also a necessary step. Given the critical importance of data when developing ML modes, we advocate that the community continues to establish proper benchmark systems and best practices for open-source data and software in order to harness the full potential of ML to improve CFD.

Acknowledgements

RV acknowledges the financial support from the Swedish Research Council (VR) and from the ERC Grant No. “2021-CoG-101043998, DEEPCONTROL”. SLB acknowledges funding support from the Army Research Office (ARO W911NF-19-1-0045; program manager Dr. Matthew Munson).

Competing interests

The authors declare no competing interests.

Author contributions

Both authors contributed equally to the ideation of the study and the writing process.

References

- [1] S. Godunov and I. Bohachevsky, “Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics,” *Matematičeskij sbornik*, vol. 47, no. 3, pp. 271–306, 1959.
- [2] R. Eymard, T. Gallouët, and R. Herbin, “Finite volume methods,” *Handbook of Numerical Analysis*, vol. 7, pp. 713–1018, 2000.
- [3] O. C. Zienkiewicz, R. L. Taylor, P. Nithiarasu, and J. Z. Zhu, *The finite element method*, vol. 3. 1977.
- [4] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral methods in fluid dynamics*. Springer Science & Business Media, 2012.
- [5] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019.
- [6] B. Recht, “A tour of reinforcement learning: The view from continuous control,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, pp. 253–279, 2019.
- [7] R. Vinuesa, H. Azizpour, I. Leite, M. Balaam, V. Dignum, S. Domisch, A. Felländer, S. D. Langhans, M. Tegmark, and F. Fuso Nerini, “The role of artificial intelligence in achieving the Sustainable Development Goals,” *Nature Communications*, vol. 11, p. 233, 2020.
- [8] F. Noé, A. Tkatchenko, K.-R. Müller, and C. Clementi, “Machine learning for molecular simulation,” *Annual review of physical chemistry*, vol. 71, pp. 361–390, 2020.
- [9] S. A. Niederer, M. S. Sacks, M. Girolami, and K. Willcox, “Scaling digital twins from the artisanal to the industrial,” *Nature Computational Science*, vol. 1, no. 5, pp. 313–320, 2021.
- [10] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal of Research and Development*, vol. 3, pp. 210–229, 1959.
- [11] M. Brenner, J. Eldredge, and J. Freund, “Perspective on machine learning for advancing fluid mechanics,” *Physical Review Fluids*, vol. 4, no. 10, p. 100501, 2019.
- [12] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, “Machine learning for fluid mechanics,” *Annual Review of Fluid Mechanics*, vol. 52, pp. 477–508, 2020.
- [13] K. Duraisamy, G. Iaccarino, and H. Xiao, “Turbulence modeling in the age of data,” *Annual Review of Fluid Mechanics*, vol. 51, pp. 357–377, 2019.
- [14] S. E. Ahmed, S. Pawar, O. San, A. Rasheed, T. Iliescu, and B. R. Noack, “On closures for reduced order models—a spectrum of first-principle to machine-learned avenues,” *Physics of Fluids*, vol. 33, no. 9, p. 091301, 2021.

- [15] B. Wang and J. Wang, "Application of artificial intelligence in computational fluid dynamics," *Industrial & Engineering Chemistry Research*, vol. 60, pp. 2772–2790, 2021.
- [16] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.
- [17] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.
- [18] F. Noé, S. Olsson, J. Köhler, and H. Wu, "Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning," *Science*, vol. 365, no. 6457, p. eaaw1147, 2019.
- [19] R. Vinuesa, S. M. Hosseini, A. Hanifi, D. S. Henningson, and P. Schlatter, "Pressure-gradient turbulent boundary layers developing around a wing section," *Flow Turbulence and Combustion*, vol. 99, pp. 613–641, 2017.
- [20] H. Choi and P. Moin, "Grid-point requirements for large eddy simulation: Chapman's estimates revisited," *Physics of Fluids*, vol. 24, p. 011702, 2012.
- [21] Y. Bar-Sinai, S. Hoyer, J. Hickey, and M. P. Brenner, "Learning data-driven discretizations for partial differential equations," *Proceedings of the National Academy of Sciences*, vol. 116, no. 31, pp. 15344–15349, 2019.
- [22] B. Stevens and T. Colonius, "Enhancement of shock-capturing methods via machine learning," *Theoretical and Computational Fluid Dynamics*, vol. 34, pp. 483–496, 2020.
- [23] J. Jeon and S. J. Kim, "FVM Network to reduce computational cost of CFD simulation," *Preprint arXiv:2105.03332*, 2021.
- [24] B. Stevens and T. Colonius, "Finitenet: A fully convolutional LSTM network architecture for time-dependent partial differential equations," *arXiv preprint arXiv:2002.03014*, 2020.
- [25] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, "Machine learning-accelerated computational fluid dynamics," *Proceedings of the National Academy of Sciences*, vol. 118, p. e2101784118, 2021.
- [26] G. J. Chandler and R. R. Kerswell, "Invariant recurrent solutions embedded in a turbulent two-dimensional Kolmogorov flow," *Journal of Fluid Mechanics*, vol. 722, pp. 554–595, 2013.
- [27] P. Bauer, A. Thorpe, and G. Brunet, "The quiet revolution of numerical weather prediction," *Nature*, vol. 525, pp. 47–55, 2015.
- [28] F. Schenk, M. Välranta, F. Muschitiello, L. Tarasov, M. Heikkilä, S. Björck, J. Brandefelt, A. V. Johansson, J. O. Näslund, and B. Wohlfarth, "Warm summers during the Younger Dryas cold reversal," *Nature Communications*, vol. 9, p. 1634, 2018.
- [29] R. Vinuesa, P. S. Negi, M. Atzori, A. Hanifi, D. S. Henningson, and P. Schlatter, "Turbulent boundary layers around wing sections up to $Re_c = 1,000,000$," *International Journal of Heat and Fluid Flow*, vol. 72, pp. 86–99, 2018.
- [30] C. Aloy Torás, P. Mimica, and M. Martínez-Sober, "Towards detecting structures in computational astrophysics plasma simulations: using machine learning for shock front classification," in *Artificial Intelligence Research and Development. Z. Falomir et al. (Eds.)*, pp. 59–63, 2018.
- [31] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier neural operator for parametric partial differential equations," *arXiv preprint arXiv:2010.08895*, 2020.
- [32] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Multipole graph neural operator for parametric partial differential equations," *arXiv preprint arXiv:2006.09535*, 2020.
- [33] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Neural operator: Graph kernel network for partial differential equations," *arXiv preprint arXiv:2003.03485*, 2020.
- [34] T. Shan, W. Tang, X. Dang, M. Li, F. Yang, S. Xu, and J. Wu, "Study on a Poisson's equation solver based on deep learning technique," *2017 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*, pp. 1–3, 2017.
- [35] Z. Zhang, L. Zhang, Z. Sun, N. Erickson, R. From, and J. Fan, "Solving Poisson's equation using deep learning in particle simulation of PN junction," *2019 Joint International Symposium on Electromagnetic Compatibility, Sapporo and Asia-Pacific International Symposium on Electromagnetic Compatibility (EMC Sapporo/APEMC)*, pp. 305–308, 2019.
- [36] R. Bridson, "Fluid simulation," *A. K. Peters, Ltd., Natick, MA, USA*, 2008.
- [37] E. Ajuria, A. Alguacil, M. Bauerheim, A. Misdariis, B. Cuenot, and E. Benazera, "Towards a hybrid computational strategy based on deep learning for incompressible flows," *AIAA AVIATION Forum, June 15–19*, pp. 1–17, 2020.
- [38] A. Özbay, A. Hamzehloo, S. Laizet, P. Tzirakis, G. Rizos, and B. Schuller, "Poisson CNN: Convolutional neural networks for the solution of the Poisson equation on a Cartesian mesh," *Data-Centric Engineering*, vol. 2, p. E6, 2021.
- [39] G. D. Weymouth, "Data-driven multi-grid solver for accelerated pressure projection," *arXiv preprint arXiv:2110.11029*, 2021.
- [40] K. Fukami, Y. Nabae, K. Kawai, and K. Fukagata, "Synthetic turbulent inflow generator using machine learning," *Physical Review Fluids*, vol. 4, p. 064603, 2019.

- [41] Y. Morita, S. Rezaeiravesh, N. Tabatabaei, R. Vinuesa, K. Fukagata, and P. Schlatter, "Applying Bayesian optimization with Gaussian-process regression to computational fluid dynamics problems," *Journal of Computational Physics*, vol. 449, p. 110788, 2022.
- [42] J. V. Boussinesq, *Théorie analytique de la chaleur: mise en harmonie avec la thermodynamique et avec la théorie mécanique de la lumière T. 2, Refroidissement et échauffement par rayonnement conductibilité des tiges, lames et masses cristallines courants de convection théorie mécanique de la lumière*. Gauthier-Villars, 1923.
- [43] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis, "CFD Vision 2030 study: a path to revolutionary computational aerospace," *Technical Report NASA/CR-2014-218178*, 2014.
- [44] J. N. Kutz, "Deep learning in fluid dynamics," *Journal of Fluid Mechanics*, vol. 814, pp. 1–4, 2017.
- [45] J. Ling, A. Kurzawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *Journal of Fluid Mechanics*, vol. 807, pp. 155–166, 2016.
- [46] T. J. Craft, B. E. Launder, and K. Suga, "Development and application of a cubic eddy-viscosity model of turbulence," *International Journal of Heat and Fluid Flow*, vol. 17, pp. 108–115, 1996.
- [47] O. Marin, R. Vinuesa, A. V. Obabko, and P. Schlatter, "Characterization of the secondary flow in hexagonal ducts," *Physics of Fluids*, vol. 28, p. 125101, 2016.
- [48] P. R. Spalart, "Strategies for turbulence modelling and simulations," *International Journal of Heat and Fluid Flow*, vol. 21, pp. 252–263, 2000.
- [49] A. Vidal, H. M. Nagib, P. Schlatter, and R. Vinuesa, "Secondary flow in spanwise-periodic in-phase sinusoidal channels," *Journal of Fluid Mechanics*, vol. 851, pp. 288–316, 2018.
- [50] J. X. Wang, J. L. Wu, and H. Xiao, "Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data," *Physical Review Fluids*, vol. 2, p. 034603, 2017.
- [51] J.-L. Wu, H. Xiao, and E. Paterson, "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework," *Physical Review Fluids*, vol. 3, p. 074602, 2018.
- [52] C. Jiang, R. Vinuesa, R. Chen, J. Mi, S. Laima, and H. Li, "An interpretable framework of data-driven turbulence modeling using deep neural networks," *Physics of Fluids*, vol. 33, p. 055133, 2021.
- [53] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, pp. 206–215, 2019.
- [54] R. Vinuesa and B. Sirmacek, "Interpretable deep-learning models to help achieve the Sustainable Development Goals," *Nature Machine Intelligence*, vol. 3, p. 926, 2021.
- [55] M. Cranmer, A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho, "Discovering symbolic models from deep learning with inductive biases," *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada. Preprint [arXiv:2006.11287](https://arxiv.org/abs/2006.11287), 2020.
- [56] J. Weatheritt and R. D. Sandberg, "A novel evolutionary algorithm applied to algebraic modifications of the RANS stress-strain relationship," *Journal of Computational Physics*, vol. 325, pp. 22–37, 2016.
- [57] J. R. Koza, "Genetic Programming: On the Programming of Computers by Means of Natural Selection," *MIT Press*, 1992.
- [58] J. Weatheritt and R. D. Sandberg, "The development of algebraic stress models using a novel evolutionary algorithm," *International Journal of Heat and Fluid Flow*, vol. 68, pp. 298–318, 2017.
- [59] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.
- [60] S. Beetham and J. Capecelatro, "Formulating turbulence closures using sparse regression with embedded form invariance," *Physical Review Fluids*, vol. 5, no. 8, p. 084611, 2020.
- [61] M. Schmelzer, R. P. Dwight, and P. Cinnella, "Discovery of algebraic reynolds-stress models using sparse symbolic regression," *Flow, Turbulence and Combustion*, vol. 104, no. 2, pp. 579–603, 2020.
- [62] S. Beetham, R. O. Fox, and J. Capecelatro, "Sparse identification of multiphase turbulence closures for coupled fluid-particle flows," *Journal of Fluid Mechanics*, vol. 914, 2021.
- [63] S. Rezaeiravesh, R. Vinuesa, and P. Schlatter, "On numerical uncertainties in scale-resolving simulations of canonical wall turbulence," *Computers and Fluids*, vol. 227, p. 105024, 2021.
- [64] M. Emory, J. Larsson, and G. Iaccarino, "Modeling of structural uncertainties in Reynolds-averaged Navier–Stokes closures," *Physics of Fluids*, vol. 25, p. 110822, 2013.
- [65] A. A. Mishra and G. Iaccarino, "Uncertainty estimation for Reynolds-averaged Navier–Stokes predictions of high-speed aircraft nozzle jets," *AIAA Journal*, vol. 55, pp. 3999–4004, 2017.
- [66] S. Poroseva, F. J. D. Colmenares, and S. Murman, "On the accuracy of RANS simulations with DNS data," *Physics of Fluids*, vol. 28, p. 115102, 2016.
- [67] 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," *Journal of Fluid Mechanics*, vol. 869, pp. 553–586, 2019.
- [68] O. Obiols-Sales, A. Vishnu, N. Malaya, and A. Chandramowliswaran, "CFDNet: a deep learning-based accelerator for fluid simulations," Preprint [arXiv:2005.04485](https://arxiv.org/abs/2005.04485), 2020.

- [69] P. Spalart and S. Allmaras, "A one-equation turbulence model for aerodynamic flows," *30th Aerospace Sciences Meeting and Exhibit, AIAA Paper 1992-0439*, 1992.
- [70] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Computers in Physics*, vol. 12, pp. 620–631, 1998.
- [71] F. Gibou, D. Hyde, and R. Fedkiw, "Sharp interface approaches and deep learning techniques for multiphase flows," *Journal of Computational Physics*, vol. 380, pp. 442–463, 2019.
- [72] M. Ma, J. Lu, and G. Tryggvason, "Using statistical learning to close two-fluid multiphase flow equations for a simple bubbly system," *Physics of Fluids*, vol. 27, p. 092101, 2015.
- [73] Y. Mi, M. Ishii, and L. H. Tsoukalas, "Flow regime identification methodology with neural networks and two-phase flow models," *Nuclear Engineering and Design*, vol. 204, pp. 87–100, 2001.
- [74] J. Smagorinsky, "General circulation experiments with the primitive equations: I. The basic experiment," *Monthly Weather Review*, vol. 91, pp. 99–164, 1963.
- [75] A. D. Beck, D. G. Flad, and C.-D. Munz, "Deep neural networks for data-driven LES closure models," *Journal of Computational Physics*, vol. 398, p. 108910, 2019.
- [76] C. J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, and T. Poinsot, "Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates," *Combustion and Flame*, vol. 203, p. 255, 2019.
- [77] R. Maulik, O. San, A. Rasheed, and P. Vedula, "Subgrid modelling for two-dimensional turbulence using neural networks," *Journal of Fluid Mechanics*, vol. 858, pp. 122–144, 2019.
- [78] R. H. Kraichnan, "Inertial ranges in two-dimensional turbulence," *Physics of Fluids*, vol. 10, pp. 1417–1423, 1967.
- [79] A. Vulliamis, G. Balarac, and C. Corre, "Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures," *Journal of Turbulence*, vol. 18, pp. 854–878, 2017.
- [80] M. Gamahara and Y. Hattori, "Searching for turbulence models by artificial neural network," *Physical Review Fluids*, vol. 2, p. 054604, 2017.
- [81] R. Maulik and O. San, "A neural network approach for the blind deconvolution of turbulent flows," *Journal of Fluid Mechanics*, vol. 831, pp. 151–181, 2017.
- [82] M. Reissmann, J. Hasslberger, R. D. Sandberg, and M. Klein, "Application of gene expression programming to a-posteriori LES modeling of a Taylor Green vortex," *Journal of Computational Physics*, vol. 424, p. 109859, 2021.
- [83] G. Novati, H. L. de Laroussilhe, and P. Koumoutsakos, "Automating turbulence modelling by multi-agent reinforcement learning," *Nature Machine Intelligence*, vol. 3, pp. 87–96, 2021.
- [84] N. Hutchins, K. Chauhan, I. Marusic, J. Monty, and J. Klewicki, "Towards reconciling the large-scale structure of turbulent boundary layers in the atmosphere and laboratory," *Boundary-Layer Meteorology*, vol. 145, pp. 273–306, 2012.
- [85] R. E. Britter and S. R. Hanna, "Flow and dispersion in urban areas," *Annual Review of Fluid Mechanics*, vol. 35, pp. 469–496, 2003.
- [86] M. G. Giometto, A. Christen, C. Meneveau, J. Fang, M. Krafczyk, and M. B. Parlange, "Spatial characteristics of roughness sublayer mean flow and turbulence over a realistic urban surface," *Boundary-Layer Meteorology*, vol. 160, pp. 425–452, 2016.
- [87] E. Bou-Zeid, C. Meneveau, and M. Parlange, "A scale-dependent Lagrangian dynamic model for large eddy simulation of complex turbulent flows," *Physics of Fluids*, vol. 17, p. 025105, 2005.
- [88] C. Moeng, "A large-eddy-simulation model for the study of planetary boundary-layer turbulence," *Journal of Atmospheric Sciences*, vol. 13, pp. 2052–2062, 1984.
- [89] Y. Mizuno and J. Jiménez, "Wall turbulence without walls," *Journal of Fluid Mechanics*, vol. 723, pp. 429–455, 2013.
- [90] M. P. Encinar, R. García-Mayoral, and J. Jiménez, "Scaling of velocity fluctuations in off-wall boundary conditions for turbulent flows," *Journal of Physics: Conference Series*, vol. 506, p. 012002, 2014.
- [91] K. Sasaki, R. Vinuesa, A. V. G. Cavalieri, P. Schlatter, and D. S. Henningson, "Transfer functions for flow predictions in wall-bounded turbulence," *Journal of Fluid Mechanics*, vol. 864, pp. 708–745, 2019.
- [92] G. B. Arivazhagan, L. Guastoni, A. Güemes, A. Ianiro, S. Discetti, P. Schlatter, H. Azizpour, and R. Vinuesa, "Predicting the near-wall region of turbulence through convolutional neural networks," *Proc. 13th ERCOFTAC Symp. on Engineering Turbulence Modelling and Measurements (ETMM13), Rhodes, Greece, September 16–17. Preprint arXiv:2107.07340*, 2021.
- [93] M. Milano and P. Koumoutsakos, "Neural network modeling for near wall turbulent flow," *Journal of Computational Physics*, vol. 182, pp. 1–26, 2002.
- [94] N. Moriya, K. Fukami, Y. Nabae, M. Morimoto, T. Nakamura, and K. Fukagata, "Inserting machine-learned virtual wall velocity for large-eddy simulation of turbulent channel flows," *Preprint arXiv:2106.09271*, 2021.
- [95] H. J. Bae and P. Koumoutsakos, "Scientific multi-agent reinforcement learning for wall-models of turbulent flows," *Nature Communications*, vol. 13, p. 1443, 2022.

- [96] K. Taira, S. L. Brunton, S. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, "Modal analysis of fluid flows: An overview," *AIAA Journal*, vol. 55, no. 12, pp. 4013–4041, 2017.
- [97] C. W. Rowley and S. T. Dawson, "Model reduction for flow analysis and control," *Annual Review of Fluid Mechanics*, vol. 49, pp. 387–417, 2017.
- [98] K. Taira, M. S. Hemati, S. L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S. Dawson, and C.-A. Yeh, "Modal analysis of fluid flows: Applications and outlook," *AIAA Journal*, vol. 58, no. 3, pp. 998–1022, 2020.
- [99] J. L. Lumley, "The structure of inhomogeneous turbulence," *Atmospheric turbulence and wave propagation*, A. M. Yaglom and V. I. Tatarski (eds). Nauka, Moscow, pp. 166–178, 1967.
- [100] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, Aug. 2010.
- [101] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks*, vol. 2, no. 1, pp. 53–58, 1989.
- [102] T. Murata, K. Fukami, and K. Fukagata, "Nonlinear mode decomposition with convolutional neural networks for fluid dynamics," *Journal of Fluid Mechanics*, vol. 882, p. A13, 2020.
- [103] H. Eivazi, S. Le Clainche, S. Hoyas, and R. Vinuesa, "Towards extraction of orthogonal and parsimonious non-linear modes from turbulent flows," *Expert Systems with Applications*, To Appear. Preprint arXiv:2109.01514, 2022.
- [104] B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele, "A hierarchy of low-dimensional models for the transient and post-transient cylinder wake," *Journal of Fluid Mechanics*, vol. 497, pp. 335–363, 2003.
- [105] K. Lee and K. T. Carlberg, "Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders," *Journal of Computational Physics*, vol. 404, p. 108973, 2020.
- [106] P. Benner, S. Gugercin, and K. Willcox, "A survey of projection-based model reduction methods for parametric dynamical systems," *SIAM Review*, vol. 57, no. 4, pp. 483–531, 2015.
- [107] K. Carlberg, M. Barone, and H. Antil, "Galerkin v. least-squares petrov–galerkin projection in nonlinear model reduction," *Journal of Computational Physics*, vol. 330, pp. 693–734, 2017.
- [108] M. Cenedese, J. Axås, B. Bäuerlein, K. Avila, and G. Haller, "Data-driven modeling and prediction of nonlinearizable dynamics via spectral submanifolds," *Nature Communications*, vol. 13, p. 872, 2022.
- [109] M. Lopez-Martin, S. Le Clainche, and B. Carro, "Model-free short-term fluid dynamics estimator with a deep 3D-convolutional neural network," *Expert Systems With Applications*, vol. 177, p. 114924, 2021.
- [110] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks," *Proceedings of the Royal Society A*, vol. 474, p. 20170844, 2018.
- [111] P. A. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, and R. Vinuesa, "Predictions of turbulent shear flows using deep neural networks," *Physical Review Fluids*, vol. 4, p. 054603, 2019.
- [112] R. Abadía-Heredia, M. López-Martín, B. Carro, J. I. Arribas, J. M. Pérez, and S. Le Clainche, "A predictive hybrid reduced order model based on proper orthogonal decomposition combined with deep learning architectures," *Expert Systems With Applications*, vol. 187, p. 115910, 2022.
- [113] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach," *Physical Review Letters*, vol. 120, no. 2, p. 024102, 2018.
- [114] E. Kaiser, B. R. Noack, L. Cordier, A. Spohn, M. Segond, M. Abel, G. Daviller, J. Osth, S. Krajnovic, and R. K. Niven, "Cluster-based reduced-order modelling of a mixing layer," *J. Fluid Mech.*, vol. 754, pp. 365–414, 2014.
- [115] B. Peherstorfer and K. Willcox, "Data-driven operator inference for nonintrusive projection-based model reduction," *Computer Methods in Applied Mechanics and Engineering*, vol. 306, pp. 196–215, 2016.
- [116] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, and K. Willcox, "Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms," *Computer Methods in Applied Mechanics and Engineering*, vol. 372, p. 113433, 2020.
- [117] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox, "Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems," *Physica D: Nonlinear Phenomena*, vol. 406, p. 132401, 2020.
- [118] J.-C. Loiseau and S. L. Brunton, "Constrained sparse Galerkin regression," *Journal of Fluid Mechanics*, vol. 838, pp. 42–67, 2018.
- [119] J.-C. Loiseau, "Data-driven modeling of the chaotic thermal convection in an annular thermosyphon," *Theoretical and Computational Fluid Dynamics*, vol. 34, no. 4, pp. 339–365, 2020.
- [120] Y. Guan, S. L. Brunton, and I. Novosselov, "Sparse nonlinear models of chaotic electroconvection," *Royal Society Open Science*, vol. 8, no. 8, p. 202367, 2021.
- [121] N. Deng, B. R. Noack, M. Morzynski, and L. R. Pastur, "Low-order model for successive bifurcations of the fluidic pinball," *Journal of Fluid Mechanics*, vol. 884, no. A37, 2020.
- [122] N. Deng, B. R. Noack, M. Morzyński, and L. R. Pastur, "Galerkin force model for transient and post-transient dynamics of the fluidic pinball," *Journal of Fluid Mechanics*, vol. 918, 2021.

- [123] J. L. Callaham, G. Rigas, J.-C. Loiseau, and S. L. Brunton, "An empirical mean-field model of symmetry-breaking in a turbulent wake," *arXiv preprint arXiv:2105.13990*, 2021.
- [124] J. L. Callaham, S. L. Brunton, and J.-C. Loiseau, "On the role of nonlinear correlations in reduced-order modeling," *arXiv preprint arXiv:2106.02409*, 2021.
- [125] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of coordinates and governing equations," *Proceedings of the National Academy of Sciences*, vol. 116, no. 45, pp. 22445–22451, 2019.
- [126] E. Yeung, S. Kundu, and N. Hodas, "Learning deep neural network representations for koopman operators of nonlinear dynamical systems," *arXiv preprint arXiv:1708.06850*, 2017.
- [127] N. Takeishi, Y. Kawahara, and T. Yairi, "Learning koopman invariant subspaces for dynamic mode decomposition," in *Advances in Neural Information Processing Systems*, pp. 1130–1140, 2017.
- [128] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature Communications*, vol. 9, no. 1, p. 4950, 2018.
- [129] A. Mardt, L. Pasquali, H. Wu, and F. Noé, "VAMPnets: Deep learning of molecular kinetics," *Nature Communications*, vol. 9, no. 5, 2018.
- [130] S. E. Otto and C. W. Rowley, "Linearly-recurrent autoencoder networks for learning dynamics," *SIAM Journal on Applied Dynamical Systems*, vol. 18, no. 1, pp. 558–593, 2019.
- [131] R. Wang, R. Walters, and R. Yu, "Incorporating symmetry into deep dynamics models for improved generalization," *arXiv preprint arXiv:2002.03061*, 2020.
- [132] R. Wang, K. Kashinath, M. Mustafa, A. Albert, and R. Yu, "Towards physics-informed deep learning for turbulent flow prediction," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1457–1466, 2020.
- [133] H. Frezat, G. Balarac, J. Le Sommer, R. Fablet, and R. Lguensat, "Physical invariance in neural networks for subgrid-scale scalar flux modeling," *Physical Review Fluids*, vol. 6, no. 2, p. 024607, 2021.
- [134] N. B. Erichson, M. Muehlebach, and M. W. Mahoney, "Physics-informed autoencoders for lyapunov-stable fluid flow prediction," *arXiv preprint arXiv:1905.10866*, 2019.
- [135] A. A. Kaptanoglu, J. L. Callaham, C. J. Hansen, A. Aravkin, and S. L. Brunton, "Promoting global stability in data-driven models of quadratic nonlinear dynamics," *Physical Review Fluids*, vol. 6, no. 094401, 2021.
- [136] R. Vinuesa, O. Lehmkuhl, A. Lozano-Durán, and J. Rabault, "Flow control in wings and discovery of novel approaches via deep reinforcement learning," *Fluids*, vol. 865, pp. 281–302, 2019.
- [137] L. Guastoni, A. Güemes, A. Ianiro, S. Discetti, P. Schlatter, H. Azizpour, and R. Vinuesa, "Convolutional-network models to predict wall-bounded turbulence from wall quantities," *Journal of Fluid Mechanics*, vol. 928, p. A27, 2021.
- [138] H. Kim, J. Kim, S. Won, and C. Lee, "Unsupervised deep learning for super-resolution reconstruction of turbulence," *Journal of Fluid Mechanics*, vol. 910, p. A29, 2021.
- [139] K. Fukami, K. Fukagata, and K. Taira, "Super-resolution reconstruction of turbulent flows with machine learning," *Journal of Fluid Mechanics*, vol. 870, pp. 106–120, 2019.
- [140] A. Güemes, S. Discetti, A. Ianiro, B. Sirmacek, H. Azizpour, and R. Vinuesa, "From coarse wall measurements to turbulent velocity fields through deep learning," *Physics of Fluids*, vol. 33, p. 075121, 2021.
- [141] K. Fukami, T. Nakamura, and K. Fukagata, "Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data," *Physics of Fluids*, vol. 32, p. 095110, 2020.
- [142] M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," *Science*, vol. 367, pp. 1026–1030, 2020.
- [143] H. Eivazi, M. Tahani, P. Schlatter, and R. Vinuesa, "Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations," *Proc. 13th ERCOFTAC Symp. on Engineering Turbulence Modelling and Measurements (ETMM13), Rhodes, Greece, September 16–17. Preprint arXiv:2107.10711*, 2021.
- [144] Y. Kim, Y. Choi, D. Widemann, and T. Zohdi, "A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder," *Journal of Computational Physics*, p. 110841, 2021.
- [145] H. Eivazi and R. Vinuesa, "Physics-informed deep-learning applications to experimental fluid mechanics," *Preprint arXiv:2203.15402*, 2022.
- [146] S. Markidis, "The old and the new: can physics-informed deep-learning replace traditional linear solvers?," *Preprint arXiv:2103.09655v2*, 2021.
- [147] J. Kim, P. Moin, and R. Moser, "Turbulence statistics in fully developed channel flow at low Reynolds number," *Journal of Fluid Mechanics*, vol. 177, pp. 133–166, 1987.
- [148] K. Fukagata, "Towards quantum computing of turbulence," *Nature Computational Science*, vol. 2, pp. 68–69, 2022.
- [149] L. A. Barba, "The hard road to reproducibility," *Science*, vol. 354, no. 6308, pp. 142–142, 2016.
- [150] O. Mesnard and L. A. Barba, "Reproducible and replicable computational fluid dynamics: it's harder than you think," *Computing in Science & Engineering*, vol. 19, no. 4, pp. 44–55, 2017.