Emerging Trends in Machine Learning for Computational Fluid Dynamics

Ricardo Vinuesa, *KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden*

Steven L. Brunton, *University of Washington, Seattle, WA, 98195, USA*

The renewed interest from the scientific community in machine learning (ML) is opening many new areas of research. Here we focus on trends in ML that are providing opportunities to advance the field of computational fluid dynamics (CFD). We discuss synergies between ML and CFD that have already shown benefits, and we also assess areas that are under development and may produce important benefits in the coming years. We believe that it is also important to emphasize a balanced perspective of cautious optimism for these emerging approaches.

Machine learning (ML) is a rapidly developing field of research that has transformed the state-of-the-art capabilities for many traditional tasks in computer science, such as image classification and captioning, natural language processing, and recommender systems. The numerous success stories of ML have led to widespread adoption in the scientific and engineering communities as well, fueled by a growing wealth of data, computing resources, and advanced optimization algorithms. This is especially true in the field of fluid mechanics, where emerging technologies complement existing computational and experimental methods, providing a unified approach to building models from data.[5]

Despite these advancements, there remains a gap in understanding how ML can be best integrated with computational fluid dynamics (CFD). This article aims to explore the synergies between ML and CFD, showcasing the potential benefits and challenges in combining these fields. ML can advance CFD in areas such as turbulence modeling, development of inflow boundary conditions, subgrid-scale (SGS) models for large-eddy simulations (LESs), closures for Reynolds(*Re*)-averaged Navier–Stokes (RANS) equations, development of reduced-order models (ROMs), and flow control.[29] Our approach is to first examine established techniques, such as proper-orthogonal decomposition (POD) and dynamic-mode decomposition (DMD), alongside deep learning techniques with autoencoders. Next, we delve into emerging opportunities where ML and CFD can be further integrated, highlighting ongoing challenges and potential solutions. We conclude by summarizing the insights gained and potential future directions for this interdisciplinary research.

OVERVIEW OF APPLICATIONS

The next sections discuss the more relevant applications at the intersection of ML and CFD.

Modeling the Near-Wall Region of Wall-Bounded Turbulence

Twenty years ago, Milano and Koumoutsakos[18] developed a method to predict the relevant features of turbulent channel flow close to the wall by using deep neural networks. In addition to the interest of this work from the perspective of modeling turbulence, and potentially developing wall models, the authors also established connections between neural networks and traditional methods, such as POD. In particular, they showed that by restricting the neural network model to be linear, the network essentially learns the features produced by standard POD. This topic has received renewed attention in recent years due to the possibility of training deeper neural network models with larger data sets as well as the emergence of novel learning architectures.[2]

Development of Inflow Conditions for Turbulence Simulations

This is a critical area for CFD when it comes to achieving high-*Re*-number conditions and simulating complex

Digital Object Identifier 10.1109/MCSE.2023.3264340
Date of current version 1 May 2023.

geometries. Spatially developing turbulent boundary layers (TBLs) requires very long domains to reach high-*Re* numbers, and if part of the low-*Re* region can be replaced by an adequate inflow condition, simulations at the relevant high-*Re* regime can become feasible. Similarly, if a simulation is designed to study the turbulent flow around a complex array of obstacles, it is beneficial to replace the inflow section of the simulation by a suitable inflow condition, thus yielding significant computational savings. Traditional approaches to these inflows have relied on various filters, recycling approaches, and imposing synthetic turbulent structures. However, recent progress in ML has enabled the use of deep learning approaches to produce robust inflow conditions, including the use of modern architectures such as transformers.[32]

> *DEEP LEARNING METHODS, HOWEVER, ARE CAPABLE OF ACCURATELY REPRODUCING THE NONLINEAR SCALE INTERACTIONS IN TURBULENT FLOWS AND HAVE RECENTLY EXHIBITED SUPERIOR PERFORMANCE COMPARED TO LINEAR METHODS.*

Development of Boundary Conditions for External-Flow Simulations

One of the challenges of simulating external flows, e.g., the flow past an airfoil, is the need for large computational domains to adequately reproduce the far-field conditions and the pressure distribution around the wing. Although the resolution of the portion of the domain near the far field is often already quite coarse, being able to replace this region entirely by a suitable boundary condition would lead to significant computational savings. Furthermore, it is important to establish prescribed pressure-gradient (PG) conditions to study PG TBLs and their nontrivial flow-history effects. To this end, a number of ML approaches have recently been developed for effective boundary-condition design.[2] In particular, Gaussian-process regression has been successfully employed to optimize the shape of the domain to prescribe a particular PG distribution.

Improved SGS Models for LESs

In this application, the goal is to develop frameworks that supplement coarse numerical simulations, where only the largest turbulent scales are resolved, with additional information corresponding to the flow scales that are too fine to be properly simulated with the chosen computational mesh.[8] This approach leads to significant computational savings but opens questions regarding the physical relevance and accuracy of the assumptions from the SGS model. One way in which ML is helping develop SGS models is using filtered, direct-numerical-simulation (DNS) data to obtain the information required to supplement the coarse model, and then to train data-driven algorithms to predict this filtered field. However, there are limitations to this approach, and recently, more nuanced approaches have been developed that replace supervised learning with reinforcement learning (RL). This also includes innovative approaches to develop wall models,[2] which can enable high-*Re*, wall-bounded turbulence studies, as discussed in more detail in the next section.

Enhanced Closures for RANS Equations

Beyond traditional approaches, based on using high-fidelity data to fit the parameters from the eddy-viscosity-based, *Re*-stress models, ML has brought a number of additional approaches to RANS modeling.[8] These approaches include the discovery of novel closure functional forms and interpretability of data-driven turbulence models. Furthermore, recent approaches based on physics-informed neural networks (PINNs)[21] have also led to successful predictions of the *Re* stresses in the context of RANS. We elaborate on these topics in the next section.

Robust Methods for Nonintrusive Sensing and Superresolution

To study and control turbulent flows experimentally, nonintrusive sensing is important, for example, being able to predict the flow based on measurements at the wall. This is an area where numerical simulations help in the design of better experiments, and where data-driven methods have traditionally played an important role. The typical methods used to predict the flow based on wall information include linear stochastic estimation, extended POD (EPOD), and system-identification approaches based on transfer functions. The main limitation of these methods is that they are linear, and although they are capable of accurately predicting the linear-superposition mechanism of turbulence, they do not capture nonlinear modulation. Deep learning methods, however, are capable of accurately reproducing the nonlinear scale interactions in turbulent flows and have recently exhibited superior performance compared to linear methods.

In particular, convolutional neural networks (CNNs), generative adversarial networks, and other computer-vision architectures provide new techniques to perform nonintrusive sensing in turbulence, even with coarsely sampled wall information.[12]

Novel Approaches for Flow Control

Finally, another area where ML is starting to have a significant impact is flow control. Numerical simulations have often been used to obtain robust control strategies, which can subsequently be deployed in experimental studies. Active flow control, in which the actuation requires energy input, has been extensively used in the context of turbulent flows, both to reduce drag and enhance mixing, depending on the application. Closed-loop control is typically the most effective approach because it uses measurements of the instantaneous state of the flow to dynamically determine the control action; note the importance of sensing mentioned previously. One traditional method of closed-loop control is opposition control, in which a synthetic jet at the wall actuates with the opposite sign of the wall-normal fluctuations in the near-wall region. ML methods are enabling more sophisticated control strategies, e.g., through genetic programming (with the advantage of providing interpretable control policies) or via deep RL, which is enabling the discovery of novel control approaches in a wide range of flows,[20] including turbulence control.[14]

In the remainder of this contribution, we focus on established techniques and emerging trends, where new applications of ML algorithms have the potential to significantly impact numerical simulations of fluid flows. After assessing the synergies between ML and CFD, we then summarize our conclusions and provide an outlook.

SYNERGIES BETWEEN ML AND CFD

The intersection of ML and CFD is a rapidly growing discipline. Therefore, we have organized the following into established methods and connections, emerging and developing research areas, and ongoing challenges and limitations.

ESTABLISHED TECHNIQUES

New approaches in ML are having a significant impact on the development of ROMs. Even the most complex flows exhibit certain patterns and dominant structures, which can be used to construct models with simpler dynamics than the original physical system, but still containing some of their essential ingredients.[3,5,22] These patterns enable the development of ROMs, which can be used to make predictions of the complete system with lower fidelity at a significantly lower computational cost. Some of the most widely used methods for ROM development rely on linear algebra. For instance, the well-known POD, also known as principal component analysis, is based on the singular value decomposition (SVD) algorithm for matrices. POD reduces the dimensionality of flow-field data by identifying the most important modes of variation in the data, which are typically associated with large-scale features in the flow. We assume that a velocity field $\boldsymbol{u}(\boldsymbol{x},t)$ can be expressed as a linear combination of the mean flow $\bar{\boldsymbol{u}}(\boldsymbol{x})$ and a set of orthogonal spatial modes $\boldsymbol{\phi}_i(\boldsymbol{x})$:

$$\mathbf{u}(\mathbf{x},t) = \bar{\mathbf{u}}(\mathbf{x}) + \sum_{i=1}^{N} a_i(t)\boldsymbol{\phi}_i(\mathbf{x}) \tag{1}$$

where $a_i(t)$ are the temporal coefficients specifying the amplitudes of the spatial modes, $\boldsymbol{x}$ are the spatial coordinates, t is the time, and boldface denotes vectors. To find the temporal coefficients $a_i(t)$ in (1), we can use the snapshots of the flow-field data $\mathbf{u}(\mathbf{x}_j,t_k)$ at discrete spatial locations $\mathbf{x}_j$ and time instances t_k. These snapshots are organized into a matrix $\boldsymbol{U} = [\boldsymbol{u}_{jk}]$, where each column corresponds to a time instant and each row corresponds to a spatial location. We then perform an SVD on the matrix $\boldsymbol{U}$, obtaining a set of orthonormal spatial modes $\boldsymbol{\phi}_i(\boldsymbol{x})$ and their corresponding singular values σ_i related to the energy contained in each mode. We keep N modes by choosing a threshold for the singular values that captures a desired percentage of the total energy in the system.

Substituting the POD into the governing equations of the fluid flow, such as the Navier–Stokes equations, gives a system of equations for the coefficients. This is known as Galerkin projection, and it involves projecting the governing equations onto the subspace spanned by the POD modes. The result is a system of ordinary differential equations for the temporal coefficients $a_i(t)$: a dynamical system representing the dynamics of the flow restricted to the low-dimensional POD subspace.

Two important properties of the POD modes are their orthogonality, implying that a particular physical feature of the system will be represented in only one mode of the expansion, and the fact that they are ranked by their energetic contribution to $\boldsymbol{u}(\boldsymbol{x},t)$, thus facilitating the truncation process. Note that (1) decomposes the spatiotemporal problem into spatial modes and their temporal dynamics so that POD is fundamentally based on a separation of variables. Thus, having

$\boldsymbol{\phi}_i(\mathbf{x})$ implies that only the temporal coefficients need to be obtained to build a ROM.

A number of ML-based approaches have been used to predict the temporal coefficients in a purely data-driven manner, including the various types of recurrent neural networks,[25] Koopman-based frameworks,[9] sparse identification of nonlinear dynamics (SINDy),[4] CNNs,[17] and even more recently, transformers.[32] These data-driven approaches have enabled the development of robust ROMs for flow cases of significant complexity.

Another data-driven approach used to identify flow features is the DMD,[23,24,28] in which the spatiotemporal data can be decomposed into Fourier-like modes with their corresponding amplitudes, frequencies, and growth rates. DMD differs from POD in several ways. Although POD decomposes the data into *spatial* modes and temporal coefficients, DMD decomposes it into *dynamic* oscillatory modes with a single frequency. DMD is a purely data-driven approach and does not require an equation, as is needed in Galerkin projection. DMD captures transient dynamics, while POD is more suited for quasi-steady flow. Both are powerful tools for reduced-order modeling, and the choice between them depends on the problem at hand.

IN DEEP LEARNING, THIS TWO-STEP PROCESS IS CALLED AN AUTOENCODER: A NETWORK THAT PROJECTS HIGH-DIMENSIONAL DATA INTO AND FROM A LOWER-DIMENSIONAL LATENT SPACE.

The methods discussed in this section learn a linear subspace where the dynamics of the ROM are expressed. Deep learning enables learning a nonlinear coordinate system on a curved manifold, where it is possible to compactly represent the complex, nonlinear dynamics of fluid-flow systems.[11] In a neural network, each layer produces an output by taking a linear combination of the inputs, adding a bias, and passing it through a nonlinear activation function. When we consider a linear neural network (i.e., the activation function is linear) with an identity mapping, we find that the matrix of weights is equivalent to the matrix formed with the eigenfunctions of POD, implying that the linear operations used in POD can be seen as the operations of a linear neural network. We can obtain the POD via training a linear neural network using backpropagation with the squared distance between the original and reconstructed data as the cost function. As discussed earlier, Milano and Koumoutsakos[18] used this idea to develop a nonlinear generalization of POD by adding a nonlinear activation function in two hidden layers of the neural network. They used this model for reconstruction of the near-wall flow in a turbulent channel, finding that it gives better compression than linear POD (better reconstruction for unseen data) at a higher computational cost.

In the nonlinear generalization of POD using a neural network, one is replacing SVD by backpropagation to learn a lower-dimensional representation of the data. This low-dimensional representation can be considered the nonlinear principal components of the data, capturing the essential features of the flow. The neural network model can then be used to reconstruct the high-dimensional, fluid-flow data from the low-dimensional representation. In deep learning, this two-step process is called an autoencoder: a network that projects high-dimensional data into and from a lower-dimensional latent space. In an autoencoder, the so-called encoder $\mathcal{E}$ maps the input data $\boldsymbol{u}$ to a low-dimensional latent space $\boldsymbol{r}$, i.e., $\boldsymbol{u} \rightarrow \boldsymbol{r}$, while the decoder $\mathcal{D}$ maps the latent space back to the original data: $\boldsymbol{r} \rightarrow \tilde{\mathbf{u}}$, obtaining the reconstruction $\tilde{\mathbf{u}}$. When the autoencoder produces an accurate reconstruction, it also provides an effective, low-dimensional representation of the original data in the latent space. The process for implementing an autoencoder is summarized as follows:

$$\mathcal{F} = \mathcal{D} \circ \mathcal{E} \tag{2a}$$

$$\tilde{\mathbf{u}} = \mathcal{F}(\mathbf{u}; \mathbf{w}) \tag{2b}$$

$$\mathcal{L}_{\text{rec}} = \varepsilon(\mathbf{u}, \tilde{\mathbf{u}}). \tag{2c}$$

Here, $\mathcal{F}$ denotes the network, $\mathbf{w}$ is its parameters to be optimized through training, ε is the loss function, and $\mathcal{L}_{\text{rec}}$ is the reconstruction loss. Through this data-driven framework, it is been possible to obtain nonlinear modal decompositions of a number of flows, leading to very compact representations of the original data in the latent space due to the inherent nonlinearity of the method.

Autoencoders, unlike POD, do not inherently possess the two most important properties of orthogonality, and rank ordering by the energy content. Recent work has addressed both issues. Obtaining the nonlinear modes in ranked order of their energy content is possible with hierarchical encoders, which leverage convolutional layers to learn increasingly abstract features.[13] They thus can exploit spatial locality and multiscale information in the flow input data. The hierarchical formulation starts by using a network, denoted as $\mathcal{F}_1$, to map the original data into a 1-D latent vector:

$\mathbf{r}_1 = \mathcal{E}_1(\mathbf{u})$. Next, a second network, $\mathcal{F}_2$, is trained to map the input data $\mathbf{u}$ into a 2-D latent space, defined as: $[\mathbf{r}_1, \mathbf{r}_2]$, where $\mathbf{r}_1$ is already known and $\mathbf{r}_2$ is obtained at this level. This recurrent relationship of networks can be extended to higher dimensions, building a ROM where each additional latent vector contributes less to the reconstruction of the original field. To promote orthogonality, disentanglement among the latent vectors can be favored by adding a β-weighted penalty term to the loss function in variational autoencoders. In tests using a complex turbulent flow,[10] this methodology provided approximately 90% energy reconstruction with just five autoencoder modes, whereas five POD modes would yield a reconstruction of roughly 30% of the energy. These results highlight the potential of using autoencoders to develop very compact ROMs, even in turbulent flows.

Established ML techniques are also playing an important role in developing improved RANS models.[8] Although traditional approaches focus on fitting coefficients in classical RANS strategies, researchers are investigating innovative methods for model discovery. These methods include genetic programming[7] or SINDy.[4] The key advantage of these approaches is their ability to provide interpretable models that are not constrained by the functional forms of existing ones. Interpretability is particularly important as many deep learning-based approaches that demonstrate potential for novel RANS models are not interpretable. Cranmer et al.[7] present a promising approach to incorporating interpretability into RANS models developed through deep learning. In their work, symbolic equations are derived through symbolic regression (using genetic programming) based on the input–output behavior of a trained neural network model. These models demonstrate performance equivalent to the original neural network while offering enhanced interpretability and generalizability properties.

ML shows great potential in other modeling tasks, such as assessing the impact of compressibility on the flow[31] and evaluating the properties of stratified flows,[27] even at large scales. Measurement techniques are also being improved through the integration of ML,[30] particularly when combined with simulations.

EMERGING OPPORTUNITIES

Large-scale simulations of turbulent flows, particularly at high *Re*, are computationally expensive due to the multiscale nature of turbulence. With increasing *Re*, the separation of scales increases and the smallest scales near the wall become progressively smaller. DNS requires computational meshes fine enough to resolve all relevant flow scales, leading to a growing number of required grid points proportional to $Re^{37/14}$ (or $Re^{13/7}$ for an LES).[6] As a result, simulating industrially relevant *Re* numbers remains infeasible with current high-performance computing facilities. Reducing the computational cost of these simulations is critically important for the field of CFD, particularly for turbulent flows in conditions close to full-scale applications.

Kochkov et al.[15] proposed an interesting approach for accelerating simulations using ML. Their method involves reducing the resolution of the numerical mesh, which automatically lowers computational cost. However, using a coarse mesh with conventional numerical methods often gives incorrect physics and erroneous results. Kochkov et al.[15] propose a deep learning framework to establish a correction between the low-cost, coarse-resolution simulation and the costly, fine-resolution simulation. Their results show that, even with computational meshes 8–10-times coarser in each spatial direction, it is still possible to recover the most relevant features of the flow, including some much smaller than the grid spacing in the coarse case, yielding excellent agreement with the full-resolution reference data. Their strategy involves filtering the fine-resolution data to produce a coarse-resolution dataset for the same time steps as the original mesh. A CNN is then trained to predict the residual between the coarse and fine simulations at each step, effectively learning how to supplement the coarse simulation with missing information. CNNs are widely used in computer vision because they can exploit spatial correlations in the data. The vortical flow used in this study exhibits coherent structures that can be effectively predicted using the various convolutional filters in the CNN. Early hidden layers focus on simpler and more abstract features, while layers closer to the output can reproduce and synthesize more complex features due to the concatenated application of convolutions of this architecture. This hierarchical representation of structures is well suited to the multiscale nature of turbulent flows, making CNNs a natural choice for such predictive tasks. In general, superresolution techniques have shown great promise in highly structured, multiscale fluid applications.[12]

It is important to note that the flow case used in Kochkov et al.[15] is relatively simple: the 2-D Kolmogorov flow sustained by an external-forcing term. Although the results in this flow case are quite promising, it is crucial to assess the feasibility of this approach in more complex scenarios, such as 3-D, unforced turbulence, where fluctuations are maintained through properly resolving the near-wall region, where most of the production occurs up to moderate *Re* numbers.

Successfully accelerating such a flow simulation would be a significant milestone in CFD and the study of high-*Re* turbulence. In a related context, the neural-operator framework[16] is a promising approach, where a generalization of neural networks is proposed to learn operators capable of approximating complex nonlinear operators. This has already been shown to perform well for coarse meshes in two dimensions at a reduced computational cost compared to traditional partial differential equation solvers.

Using coarser meshes and supplementing the resolved flow with an ML-based term to recover unresolved information can also be combined with a wall model. In this approach, a model is used to calculate the wall-shear stress, eliminating the need to resolve the computationally expensive area close to the wall, and yielding significant computational savings, particularly at a higher *Re* number. This approach is popular in atmospheric boundary layers,[26] where extreme flow conditions typically require such an approach for realistic *Re* numbers. An alternative method involves exploiting the self-similarity of turbulent structures within the overlap region for a more sophisticated condition.[19] Although promising, current wall models have limitations and can benefit from ML to devise more advanced strategies. Some studies use CNNs[1] to predict fluctuations at the boundary-condition plane based on information from another plane. Even though these methods show better predictions, missing information and scales at the input plane hinder the development of robust models that perform well a posteri. An interesting direction involves using RL to predict the most likely output plane based on the input plane, leading to adequate flow features and turbulence statistics.[2] The advantage of this approach is its unsupervised nature, which does not require labeled data for the boundary condition. The most suitable boundary condition may not be the filtered DNS plane; thus, exploring alternatives given a predefined reward function could be a viable approach for these simulations.

ONGOING CHALLENGES

Several aspects of CFD are unlikely to be easily improved or replaced by ML. Many of the numerical methods underpinning CFD have been optimized for decades and provide incredible scalability and tunable accuracy. Imagining that ML will replace these classical numerical algorithms is unrealistic, and a more practical goal is to design new algorithms and frameworks that allow them to work together.

Further, ML models are generally quite expensive to train, both in terms of the learning algorithm itself and the cost of generating the training data. Training data are typically generated using high-fidelity CFD, and this cost is often neglected when assessing the performance of the overall techniques, compared with traditional methods. We should also be mindful that we are comparing emerging ML techniques, many of which are fewer than 10 years old, with mature numerical algorithms that have been tested and developed for over half a century. Nevertheless, it is important to include these training costs and clarify how these models will be incorporated into existing workflows.

A number of important questions arise when assessing an ML-based CFD solution. How general is the model? Does it only apply to the parameters explored in the training process, or does it generalize? How often will the model be used, and what fidelity is required for the ultimate objective? For example, the fidelity required for closed-loop feedback control may be considerably lower than that which is required to optimize an airfoil geometry with transonic separation. The answers to these questions inform the higher-level questions of how ML solutions will be incorporated into existing CFD workflows, and how training costs should be quantified and balanced against online acceleration. The community is still largely in the *basic research* phase of incorporating ML into scientific computing as it remains unclear exactly how these algorithms will be used in production. Therefore, it is important to be as precise and transparent as possible about the strengths and weaknesses of these approaches.

ML solutions, especially approaches based on deep learning, typically require either large volumes of data or are limited to a narrow set of parameters explored in the training process. Improvements in transfer learning may make it possible to extend models trained at one set of parameters to another parameter set, for example, generalizing an ML model to higher *Re* numbers; however, this capability is currently lacking. In CFD applications, where acquiring training data requires expensive simulations, ML will likely benefit from active learning approaches that are thoughtful in selecting training data to reduce uncertainty and improve model performance in a targeted way. Many parallels exist among enhancing CFD with ML, the history of enhancing experiments with CFD, and enhancing CFD with ROMs. We can leverage this vast literature on uncertainty quantification, design of experiments, variable-fidelity methods, and targeted idealizations to guide ML solutions.

A promising approach to reducing training costs while improving generalizability is to incorporate known physics into the ML models. In fact, a good working definition of *physics* is models that generalize to new, unseen

FROM THE EDITOR-IN-CHIEF

Recent years have seen a whirlwind of new research applications of machine learning in science and engineering. In the eye of the vortex, calm contemplation of the opportunities and pending deficiencies are still rare. This illuminated outlook by Ricardo Vinuesa and Steven L. Brunton fills that gap, giving us a refreshing mix of excitement and balance in an accessible form. It is recommended reading for all the members of your team!

Lorena A. Barba
Editor-in-Chief

scenarios. Emerging techniques, such as PINNs,[21] incorporate known physics into ML techniques, for example, by including a loss function to promote the governing equation being satisfied. Although these approaches often provide improvements over training algorithms that do not incorporate any physical knowledge, adding terms to the loss function essentially acts as a *suggestion* that the solution is physical. This is in contrast to numerical techniques that are structurally formulated or constrained to enforce physics up to a specified numerical tolerance. Similarly, we should balance how much prior physical knowledge is enforced, and how rigidly, with the flexibility to capture new effects, such as model discrepancies and inadequacies. In the limit of training with infinite data from high-fidelity, Navier–Stokes simulations, will an ML model converge to the simulation itself? Will it converge to the Navier–Stokes equations? These are fundamental open questions in ML, and incorporating physics is an area of intense active research.

CONCLUSION

Several existing, emerging, and ongoing connections are found between ML and CFD. Some challenges in CFD are particularly amenable to ML solutions, such as developing ROMs for large, energetic, coherent structures and improved RANS closure models in turbulence. In these areas, large volumes of data, often from CFD, are used to extract patterns and simplified models that may be used to accelerate future simulations, balancing accuracy and computational cost. These fields benefit from decades of experience in incorporating data-driven models and managing uncertainties through multifidelity methods and design of experiments. Thus, it is natural to incorporate ML solutions into this framework.

Other aspects of CFD are promising for augmentation through ML, although developments appear to be more challenging and improvements may be more modest. For example, CFD may be accelerated by learning-correction terms to accurately capture high-fidelity physics on a coarser mesh, followed by a super-resolution map from the coarse mesh back to the fine mesh. Although initial demonstrations have been quite promising, a number of ongoing questions and challenges remain before these approaches are ready for full-scale integration with existing CFD techniques. Current demonstrations typically involve 2-D flows, and significant developments will likely be needed to extend these to more realistic 3-D flow configurations. Mature CFD techniques, such as spectral-element and finite-volume methods, are well developed, and they have been optimized for several decades to have a low-memory footprint, and to scale to extremely large problem sizes through parallel processing. It may be unreasonable to expect emerging ML solutions to compete with these mature techniques without similar concerted research efforts, and time will tell how much benefit will be achieved. Open questions remain about the generalizability of ML models and how they will be incorporated into existing workflows, making it difficult to provide a holistic comparison of various approaches.

Similarly, most of these techniques require extensive training databases, which are typically based on the high-fidelity CFD solutions that they aim to enhance/augment/replace. The cost of generating the training data is often considered a *one-time, offline* cost, with the hope of future online computations being accelerated. However, depending on the generalization capabilities of the resulting model and the number of future evaluations, the training cost may be significant and should be considered in a holistic assessment of the algorithm. The offline/online split is especially valid for ROMs that will be used for control applications, where low-latency predictions are often more important than model fidelity. ROM literature also has extensive experience with quantifying the trade-offs between training cost, model fidelity, and online costs, for example, to maximize the use of limited computational resources to iteratively optimize airfoil geometries for Formula One cars. This literature, which involves uncertainty quantification and iterative design of experiments, should provide a useful basis for the challenging task of incorporating ML into CFD workflows. We also have opportunities to improve optimization workflows, using automatic differentiation in ML frameworks to replace costly adjoint calculations.

It is important to balance optimism for the potential of ML to enhance CFD with a healthy respect for how well existing CFD works. We see a strong parallel between the current rise of ML and the rise of CFD in the 1980s. In retrospect, CFD was never destined to replace experiments as it is a complementary approach that offers different strengths and weaknesses. Similarly, it should be clear that ML will not replace CFD; rather, it will complement and enhance our capabilities, providing another approach to develop improved models based on available data. The rise of ML techniques in computational science, and in particular, in CFD, will require educational efforts to incorporate these skills into the standard arsenal of graduating engineers. Just as engineering students are now expected to have computational proficiency, future students will be expected to have basic proficiency in ML.

ACKNOWLEDGMENTS

Ricardo Vinuesa acknowledges financial support from the European Research Council (ERC) Grant 2021-CoG-101043998 "DEEPCONTROL." Steven L. Brunton acknowledges funding support from the U.S. Army Research Office (ARO W911NF-19-1-0045) and the National Science Foundation AI Institute in Dynamic Systems (Grant 2112085).

REFERENCES

1. G. B. Arivazhagan et al., "Predicting the near-wall region of turbulence through convolutional neural networks," in *Proc. 13th ERCOFTAC Symp. Eng. Turbulence Model. Meas. (ETMM13)*, Rhodes, Greece, 2021, pp. 16–17.
2. H. J. Bae and P. Koumoutsakos, "Scientific multi-agent reinforcement learning for wall-models of turbulent flows," *Nature Commun.*, vol. 13, no. 1, Mar. 2022, Art. no. 1443, doi: 10.1038/s41467-022-28957-7.
3. P. Benner, S. Gugercin, and K. Willcox, "A survey of projection-based model reduction methods for parametric dynamical systems," *SIAM Rev.*, vol. 57, no. 4, pp. 483–531, 2015, doi: 10.1137/130932715.
4. S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 15, pp. 3932–3937, Apr. 2016, doi: 10.1073/pnas.1517384113.
5. S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," *Annu. Rev. Fluid Mech.*, vol. 52, no. 1, pp. 477–508, Jan. 2020, doi: 10.1146/annurev-fluid-010719-060214.
6. H. Choi and P. Moin, "Grid-point requirements for large eddy simulation: Chapman's estimates revisited," *Phys. Fluids*, vol. 24, no. 1, Jan. 2012, Art. no. 011702, doi: 10.1063/1.3676783.
7. M. Cranmer et al., "Discovering symbolic models from deep learning with inductive biases," in *Proc. Adv. Neural Inf. Process. Syst. 33*, 2020, pp. 17,429–17,442.
8. K. Duraisamy, G. Iaccarino, and H. Xiao, "Turbulence modeling in the age of data," *Annu. Rev. Fluid Mech.*, vol. 51, no. 1, pp. 357–377, Jan. 2019, doi: 10.1146/annurev-fluid-010518-040547.
9. H. Eivazi, L. Guastoni, P. Schlatter, H. Azizpour, and R. Vinuesa, "Recurrent neural networks and Koopman-based frameworks for temporal predictions in a low-order model of turbulence," *Int. J. Heat Fluid Flow*, vol. 90, Aug. 2021, Art. no. 108816, doi: 10.1016/j.ijheatfluidflow.2021.108816.
10. H. Eivazi, S. Le Clainche, S. Hoyas, and R. Vinuesa, "Towards extraction of orthogonal and parsimonious non-linear modes from turbulent flows," *Expert Syst. Appl.*, vol. 202, Sep. 2022, Art. no. 117038, doi: 10.1016/j.eswa.2022.117038.
11. D. Floryan and M. Graham, "Data-driven discovery of intrinsic dynamics," *Nature Mach. Intell.*, vol. 4, no. 12, pp. 1113–1120, Dec. 2022, doi: 10.1038/s42256-022-00575-4.
12. K. Fukami, K. Fukagata, and K. Taira, "Super-resolution reconstruction of turbulent flows with machine learning," *J. Fluid Mechanics*, vol. 870, pp. 106–120, Jul. 2019, doi: 10.1017/jfm.2019.238.
13. K. Fukami, T. Nakamura, and K. Fukagata, "Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data," *Phys. Fluids*, vol. 32, no. 9, Sep. 2020, Art. no. 095110, doi: 10.1063/5.0020721.
14. L. Guastoni, J. Rabault, P. Schlatter, H. Azizpour, and R. Vinuesa, "Deep reinforcement learning for turbulent drag reduction in channel flows," 2023, *arXiv:2301.09889*.
15. D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, "Machine learning-accelerated computational fluid dynamics," *Proc. Nat. Acad. Sci. USA*, vol. 118, no. 21, May 2021, Art. no. e2101784118, doi: 10.1073/pnas.2101784118.
16. Z. Li et al., "Neural operator: Graph kernel network for partial differential equations," 2020, *arXiv:2003.03485*.
17. M. Lopez-Martin, S. Le Clainche, and B. Carro, "Model-free short-term fluid dynamics estimator with a deep 3D-convolutional neural network," *Expert Syst. Appl.*, vol. 177, Sep. 2021, Art. no. 114924, doi: 10.1016/j.eswa.2021.114924.
18. M. Milano and P. Koumoutsakos, "Neural network modeling for near wall turbulent flow," *J. Comput.*

Phys., vol. 182, no. 1, pp. 1–26, Oct. 2002, doi: 10.1006/jcph.2002.7146.

19. Y. Mizuno and J. Jiménez, "Wall turbulence without walls," *J. Fluid Mechanics*, vol. 723, pp. 429–455, May 2013, doi: 10.1017/jfm.2013.137.
20. J. Rabault, M. Kuchta, A. Jensen, U. Réglade, and N. Cerardi, "Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control," *J. Fluid Mechanics*, vol. 865, pp. 281–302, Apr. 2019, doi: 10.1017/jfm.2019.62.
21. M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019, doi: 10.1016/j.jcp.2018.10.045.
22. C. W. Rowley and S. T. Dawson, "Model reduction for flow analysis and control," *Annu. Rev. Fluid Mech.*, vol. 49, no. 1, pp. 387–417, Jan. 2017, doi: 10.1146/annurev-fluid-010816-060042.
23. C. W. Rowley, I. Mezic, S. Bagheri, P. Schlatter, and D. Henningson, "Spectral analysis of nonlinear flows," *J. Fluid Mechanics*, vol. 641, pp. 115–127, Dec. 2009, doi: 10.1017/S0022112009992059.
24. P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *J. Fluid Mechanics*, vol. 656, pp. 5–28, Aug. 2010, doi: 10.1017/S0022112010001217.
25. P. A. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, and R. Vinuesa, "Predictions of turbulent shear flows using deep neural networks," *Physical Rev. Fluids*, vol. 4, no. 5, May 2019, Art. no. 054603, doi: 10.1103/PhysRevFluids.4.054603.
26. R. Stoll, J. A. Gibbs, S. T. Salesky, W. Anderson, and M. Calaf, "Large-eddy simulation of the atmospheric boundary layer," *Boundary-Layer Meteorol.*, vol. 177, nos. 2–3, pp. 541–581, Dec. 2020, doi: 10.1007/s10546-020-00556-3.
27. C. Tan, F. Li, S. Lv, Y. Yang, and F. Dong, "Gas-liquid two-phase stratified flow interface reconstruction with sparse batch normalization convolutional neural network," *IEEE Sensors J.*, vol. 21, no. 15, pp. 17,076–17,084, Aug. 2021, doi: 10.1109/JSEN.2021.3081432.
28. J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, "On dynamic mode decomposition: Theory and applications," *J. Comput. Dyn.*, vol. 1, no. 2, pp. 391–421, 2014, doi: 10.3934/jcd.2014.1.391.
29. R. Vinuesa and S. L. Brunton, "Enhancing computational fluid dynamics with machine learning," *Nature Comput. Sci.*, vol. 2, no. 6, pp. 358–366, Jun. 2022, doi: 10.1038/s43588-022-00264-7.
30. R. Vinuesa, S. L. Brunton, and B. J. McKeon, "The transformative potential of machine learning for experiments in fluid mechanics," 2023, *arXiv:2303.15832*.
31. J.-X. Wang, J. Huang, L. Duan, and H. Xiao, "Prediction of Reynolds stresses in high-Mach-number turbulent boundary layers using physics-informed machine learning," *Theor. Comput. Fluid Dyn.*, vol. 33, no. 1, pp. 1–19, Feb. 2019, doi: 10.1007/s00162-018-0480-2.
32. M. Z. Yousif, M. Zhang, L. Yu, R. Vinuesa, and H. Lim, "A transformer-based synthetic-inflow generator for spatially developing turbulent boundary layers," *J. Fluid Mechanics*, vol. 957, Feb. 2023, Art. no. A6, doi: 10.1017/jfm.2022.1088.

RICARDO VINUESA is an associate professor with the Department of Engineering Mechanics, KTH Royal Institute of Technology, SE-100 44, Stockholm, Sweden. He is also vice director of the KTH Digitalization Platform and lead faculty at the KTH Climate Action Centre. His research interests include machine learning, high-performance computing, turbulent flows, and flow control. Vinuesa received his Ph.D. degree in mechanical and aerospace engineering from the Illinois Institute of Technology. He is a member of the Swedish e-Science Research Centre. He is the corresponding author of this article. Contact him at rvinuesa@mech.kth.se.

STEVEN L. BRUNTON is a professor with the Department of Mechanical Engineering at the University of Washington, Seattle, WA, 98185, USA. He is also associate director of the AI Institute in Dynamic Systems. His research interests include machine learning, fluid dynamics, dynamical systems, and control. Brunton received his Ph.D. degree in mechanical and aerospace engineering from Princeton University. Contact him at sbrunton@uw.edu.