Acta Mechanica Sinica (2021) 37(12):1718–1726
https://doi.org/10.1007/s10409-021-01143-6

INVITED REVIEW

Applying machine learning to study fluid mechanics

Steven L. Brunton[1]

Received: 24 August 2021 / Revised: 8 September 2021 / Accepted: 16 September 2021 / Published online: 4 January 2022

Abstract
This paper provides a short overview of how to use machine learning to build data-driven models in fluid mechanics. The process of machine learning is broken down into five stages: (1) formulating a problem to model, (2) collecting and curating training data to inform the model, (3) choosing an architecture with which to represent the model, (4) designing a loss function to assess the performance of the model, and (5) selecting and implementing an optimization algorithm to train the model. At each stage, we discuss how prior physical knowledge may be embedding into the process, with specific examples from the field of fluid mechanics.

Keywords Machine learning · Fluid mechanics · Physics-informed machine learning · Neural networks · Deep learning

1 Introduction

The field of fluid mechanics is rich with data and rife with problems, which is to say that it is a perfect playground for machine learning. Machine learning is the art of building models from data using optimization and regression algorithms. Many of the challenges in fluid mechanics may be posed as optimization problems, such designing a wing to maximize lift while minimizing drag at cruise velocities, estimating a flow field from limited measurements, controlling turbulence for mixing enhancement in a chemical plant or drag reduction behind a vehicle, among myriad others. These optimization tasks fit well with machine learning algorithms, which are designed to handle nonlinear and high-dimensional problems. In fact, machine learning and fluid mechanics both tend to rely on the same assumption that there are patterns that can be exploited, even in high-dimensional systems [1]. Often, the machine learning algorithm will model some aspect of the fluid, such as the lift profile given a particular airfoil geometry, providing a *surrogate* that may be optimized over. Machine learning may also be used to directly solve the fluid optimization task, such as designing a machine learning model to manipulate the behavior of the fluid for some engineering objective with active control [2–4].

In either case, it is important to realize that machine learning is *not* an automatic or turn-key procedure for extracting models from data. Instead, it requires expert human guidance at every stage of the process, from deciding on the problem, to collecting and curating data that might inform the model, to selecting the machine learning architecture best capable of representing or modeling the data, to designing custom loss functions to quantify performance and guide the optimization, to implementing specific optimization algorithms to train the machine learning model to minimize the loss function over the data. A better name for machine learning might be "expert humans teaching machines how to learn a model to fit some data," although this is not as catchy. Particularly skilled (or lucky!) experts may design a learner or a learning framework that is capable of learning a variety of tasks, generalizing beyond the training data, and mimicking other aspects of intelligence. However, such artificial intelligence is rare, even more so than human intelligence. The majority of machine learning models are just that, models, which should fit directly into the decades old practice of model-based design, optimization, and control [5].

With its unprecedented success on many challenging problems in computer vision and natural language processing, machine learning is rapidly entering the physical sciences, and fluid mechanics is no exception. The simultaneous promise, and over-promise, of machine learning is causing many researchers to have a healthy mixture of optimism and

Executive Editor: Bernd Noack.

✉ Steven L. Brunton
sbrunton@uw.edu

[1] Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, USA

skepticism. In both cases, there is a strong desire to understand the uses and limitations of machine learning, as well as best practices for how to incorporate it into existing research and development workflows. It is also important to realize that while it is now relatively simple to train a machine learning model for a well-defined task, it is still quite difficult to create a new model that outperforms traditional numerical algorithms and physics-based models. Incorporating partially known physics into the machine learning pipeline well tend to improve model generalization and improve interpretability and explainability, which are key elements of modern machine learning [6,7].

2 Physics informed machine learning for fluid mechanics

Applied machine learning may be separated into a few canonical steps, each of which provides an opportunity to embed prior physical knowledge: (1) choosing the problem to model or the question to answer; (2) choosing and curating the data used to train the model; (3) deciding on a machine learning architecture to best represent or model this data; (4) designing loss functions to quantify performance and to guide the learning process; and (5) implementing an optimization algorithm to train the model to minimize the loss function over the training data. See Fig. 1 for a schematic of this process on the example of reduced-order modeling. This organization of steps is only approximate, and there are considerable overlaps and tight interconnections between each stage. For example, choosing the problem to model and choosing the data to inform this model are two closely related decisions. Similarly, designing a custom loss function and implementing an optimization algorithm to minimize this loss function are tightly coupled. In most modern machine learning workflows, it is common to iteratively revisit earlier stages based on the outcome at later stages, so that the machine learning researcher is constantly asking new questions and revising the data, the architecture, the loss functions, and the optimization algorithm to improve performance. Here, we discuss these canonical stages of machine learning, investigate how to incorporate physics, and review examples in the field of fluid mechanics. This discussion is largely meant to be a high-level overview, and many more details can be found in recent reviews [5,8–10].

2.1 The problem

Data science is the art of asking and answering questions with data. The sub-field of machine learning is concerned with leveraging historical data to build models that may be deployed to automatically answer these questions, ideally in real-time, given new data. It is critical to select the right system to model, motivated by a problem that is both important and tractable. Choosing a problem involves deciding on input data that will be readily available in the future, and output data that will represent the desired output, or prediction, of the model. The output data should be determinable from the inputs, and the functional relationship between these is precisely what the machine learning model will be trained to capture.

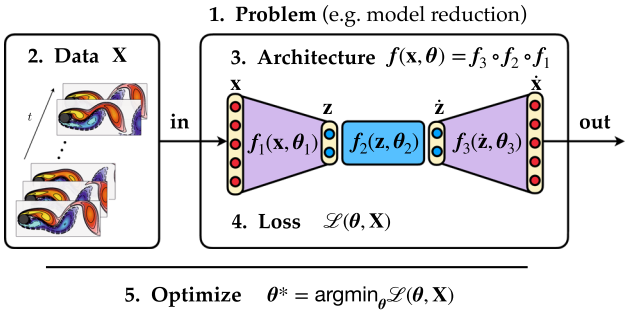

Fig. 1 Schematic of the five stages of machine learning on an example of reduced-order modeling. In this case, the goal is to learn a low dimensional coordinate system $\mathbf{z} = f_1(\mathbf{x}, \boldsymbol{\theta}_1)$ from data in a high-dimensional representation $\mathbf{x}$, along with a dynamical system model $\dot{\mathbf{z}} = f_2(\mathbf{z}, \boldsymbol{\theta}_2)$ for how the state $\mathbf{z}$ evolves in time. Finally, this latent state derivative $\dot{\mathbf{z}}$ must be able to approximate the high dimensional derivative $\dot{\mathbf{x}}$ through the decoder $\dot{\mathbf{x}} \approx f_3(\dot{\mathbf{z}}, \boldsymbol{\theta}_3)$. The loss function $\mathscr{L}(\boldsymbol{\theta}, \mathbf{X})$ defines how well the model performs, averaged over the data $\mathbf{X}$. Finally, the parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3\}$ are found through optimization

The nature of the problem, specifically what outputs will be modeled given what inputs, determines the large classes of machine learning algorithms: *supervised*, *unsupervised*, and *reinforcement learning*. In supervised learning, the training data will have expert labels that should be predicted or modeled with the machine learning algorithm. These output labels may be discrete, such as a categorical label of a "dog" or a "cat" given an input image, in which case the task is one of *classification*. If the labels are continuous, such as the average value of lift or drag given a specified airfoil geometry, then the task is one of *regression*. In unsupervised learning, there are no expert labels, and structure must be extracted from the input data alone; thus, this is often referred to as *data mining*, and constitutes a particularly challenging field of machine learning. Again, if the structure in the data is assumed to be discrete, then the task is *clustering*. After the clusters are identified and characterized, these groupings may be used as proxy labels to then classify new data. If the structure in the data is assumed to be continuously varying, then the task is typically thought of as an *embedding* or *dimensionality reduction* task. Principal component analysis (PCA) or proper orthogonal decomposition (POD) may be thought of as unsupervised learning tasks that seek a continuous embedding of reduced dimension [11]. Reinforcement learning is a third, large branch of machine learning research, in which an *agent* learns to make control decisions to interact with an

environment for some high level objection [12]. Examples include learning how to play games [13,14], such as chess and go.

2.1.1 Embedding physics

Deciding on what phenomena to model with machine learning is often inherently related to the underlying physics. Although classical machine learning has been largely applied to "static" tasks, such as image classification and the placement of advertisements, increasingly it is possible to apply these techniques to model physical systems that evolve in time according to some *rules* or *physics*. For example, we may formulate a learning problem to find and represent a conserved quantity, such as a Hamiltonian, purely from data [15]. Alternatively, the machine learning task may be to model time-series data as a differential equation, with the learning algorithm representing the dynamical system [16–20]. Similarly, the task may involve learning a coordinate transformation where these dynamics become simplified in some *physical* way; i.e., coordinate transformations to linearize or diagonalize/decouple dynamics [21–28].

2.1.2 Examples in fluid mechanics

There are many *physical* modeling tasks in fluid mechanics that are benefiting from machine learning [5,9]. A large field of study focuses on formulating turbulence closure modeling as a machine learning problem [8,29], such as learning models for the Reynolds stresses [30,31] or sub-gridscale turbulence [32,33]. Designing useful input features is also an important way that prior physical knowledge is incorporated into turbulence closure modeling [34–36]. Similarly, machine learning has recently been focused on the problem of improving computational fluid dynamics (CFD) solvers [37–40]. Other important problems in fluid mechanics that benefit from machine learning include super-resolution [41,42], robust modal decompositions [1,43,44], network and cluster modeling [45–47], control [4,48] and reinforcement learning [49,50], and design of experiments in cyberphysical systems [51]. Aerodynamics is a large related field with significant data-driven advances [52]. The very nature of these problems embeds the learning process into a larger physics-based framework, so that the models are more physically relevant by construction.

2.2 The data

Data is the lifeblood of machine learning, and our ability to build effective models relies on what data is available or may be collected. As discussed earlier, choosing data to inform a model is closely related to choosing what to model in the first place, and therefore this stage cannot be strictly separated from the choice of a problem above. The quality and quantity of data directly affects the resulting machine learning model. Many machine learning architectures, such as deep neural networks, are essentially sophisticated interpolation engines, and so having a diversity of training data is essential to these models being useful on unseen data. For example, modern deep convolutional neural networks rose to prominence with their unprecedented classification accuracy [53] on the ImageNet data base [54], which contains over 14 million labeled images with over 20,000 categories, providing a sufficiently large and rich set of examples for training. This pairing of a vast labeled data set with a novel deep learning architecture is widely regarded as the beginning of the modern era of deep learning [55].

2.2.1 Embedding physics

The training data provides several opportunities to embed prior physical knowledge. If a system is known to exhibit a symmetry, such translational or rotational invariance, then it is possible to augment and enrich the training data with shifted or rotated examples. More generally, it is often assumed that with an abundance of training data, these physical invariances will automatically be learned by a sufficiently expressive architecture. However, this approach tends to require considerable resources, both to collect and curate the data, as well as to train increasingly large models, making it more appropriate for industrial scale, rather than academic scale, research. In contrast, it is also possible to use physical intuition to craft new features from the training data, for example by applying a coordinate transformation that may simplify the representation or training. Physical data often comes from multiple sources with different fidelity, such as from numerical simulations, laboratory experiments, and in-flight tests. This is an important area of research for flight testing and unsteady aerodynamics [52], and recently physics informed neural networks have been used with multifidelity data to approximate PDEs [56].

2.2.2 Examples in fluid mechanics

Fluids data is notoriously vast and high-dimensional, with individual flow fields often requiring millions (or more!) degrees of freedom to characterize. Moreover, these flow fields typically evolve in time, resulting in a time series of multiple snapshots. Although vast in the spatial and/or temporal dimensions, data is often rather sparse in parameter space, as it is expensive to numerically or experimentally investigate multiple geometries, Reynolds numbers, etc. Thus there are many algorithms designed for both rich and sparse data. Other considerations involve exciting transients and observing how the system evolves when it is away from its natural state. In many other cases, fluids data might be

quite limited, for example given by time-series data from a few pressure measurements on the surface of an airfoil, or from force recordings on an experimental turbine.

2.3 The architecture

Once a problem has been identified, and data is collected and curated, it is necessary to choose an architecture with which to represent the machine learning model. Typically, a machine learning model is a function that maps inputs to outputs

$$\mathbf{y} = \mathbf{f}(\mathbf{x}; \boldsymbol{\theta}) \tag{1}$$

and this function is generally represented within a specified family of functions parameterized by values in $\boldsymbol{\theta}$. For example, a linear regression model would model outputs as a linear function of the inputs, with $\boldsymbol{\theta}$ parameterizing this linear map, or matrix. Neural networks have emerged as a particularly powerful and flexible class of models to represent functional relationships between data, and they have been shown to be able to approximate arbitrarily complex functions with sufficient data and depth [57,58]. There is a tremendous variety of potential neural network architectures [11], limited only by the imagination of the human designer. The most common architecture is a simple feedforward network, in which data enters through an input layer and maps sequentially through a number of computational layers until an output layer. Each layer consists of nodes, where data from nodes in the previous layer are combined in a weighted sum and processed through an activation function, which is typically nonlinear. In this way, neural networks are fundamentally compositional in nature. The parameters $\boldsymbol{\theta}$ determine the network weights for how data is passed from one layer to the next, i.e. the weighted connectivity matrices for how nodes are connected in adjacent layers. The overarching network topology (i.e., how many layers, how large, what type of activation functions, etc.) is specified by the architect or determined in a meta-optimization, thus determining the family of functions that may be approximated by that class of network. Then, the network weights for the specific architecture are optimized over the data to minimize a given loss function; these stages are described next.

It is important to note that not all machine learning architectures are neural networks, although they are one of the most powerful and expressive modern architectures, powered by increasingly big data and high performance computing. Before the success of deep convolutional networks on the ImageNet dataset, neural networks were not even mentioned in the list of top ten machine learning algorithms [59]. Random forests [60] and support vector machines [61] are two other leading architectures for supervised learning. Bayesian methods are also widely used, especially for dynamical systems [62]. Genetic programming has also been widely used to learn human-interpretable, yet flexible representations of data for modeling [16,63–65] and control [4]. In addition, standard linear regression and generalized linear regression are still widely used for modeling time-series data, especially in fluids. The dynamic mode decomposition (DMD) [1,17,66] employs linear regression with a low-rank constraint in the optimization to find dominant spatiotemporal coherent structures that evolve linearly in time. The sparse identification of nonlinear dynamics (SINDy) [18] algorithm employs generalized linear regression, with either a sparsity promoting loss function [67] or a sparse optimization algorithm [18,68], to identify a differential equation model with as few model terms as are necessary to fit the data.

2.3.1 Embedding physics

Choosing a machine learning architecture with which to model the training data is one of the most intriguing opportunities to embed physical knowledge into the learning process. Among the simplest choices are convolutional networks for translationally invariant systems, and recurrent networks, such as long-short-time memory (LSTM) networks [20] or reservoir computing [19,69], for systems that evolve in time. LSTMs have recently been used to predict aeroelastic responses across a range of Mach numbers [70]. More generally, equivariant networks seek to encode various symmetries by construction, which should improve accuracy and reduce data requirements for physical systems [71–74]. Autoencoder networks enforce the physical notion that there should be low-dimensional structure, even for high-dimensional data, by imposing an information bottleneck, given by a constriction of the number of nodes in one or more layers of the network. Such networks uncover nonlinear manifolds where the data is compactly represented, generalizing the linear dimensionality reduction obtained by PCA and POD. It is also possible to embed physics more directly into the architecture, for example by incorporating Hamiltonian [75,76] or Lagrangian [77,78] structure. There are numerous successful examples of physics-informed neural networks (PINNs) [79–83], which solve supervised learning problems while being constrained to satisfy a governing physical law. Graph neural networks have also shown the ability to learn generalizable physics in a range of challenging domains [64,84,85]. Deep operator networks [86] are able to learn continuous operators, such as governing partial differential equations, from relatively limited training data.

2.3.2 Examples in fluid mechanics

There are numerous examples of custom neural network architectures being used to enforce physical solutions for applications in fluid mechanics. The work of Ling et al.

[30] designed a custom neural network layer that enforced Galilean invariance in the Reynolds stress tensors that they were modeling. Related Reynolds stress models have been developed using the SINDy sparse modeling approach [87–89]. Hybrid models that combine linear system identification and nonlinear neural networks have been used to model complex aeroelastic systems [90]. The hidden fluid mechanics (HFM) approach is a physics-informed neural network strategy that encodes the Navier–Stokes equations while being flexible to the boundary conditions and geometry of the problem, enabling impressive physically quantifiable flow field estimations from limited data [91]. Sparse sensing has also been used to recover pressure distributions around airfoils [92]. The Fourier neural operator is a novel operator network that performs super-resolution upscaling and simulation modeling tasks [93]. Equivariant convolutional networks have been designed and applied to enforce symmetries in high-dimensional complex systems from fluid dynamics [73]. Physical invariances have also been incorporated into neural networks for subgrid-scale scalar flux modeling [94]. Lee and Carlberg [95] recently showed how to incorporate deep convolutional autoencoder networks into the broader reduced-order modeling framework [96–98], taking advantage of the superior dimensionality reduction capabilities of deep autoencoders.

2.4 The loss function

The loss function is how we quantify how well the model is performing, often on a variety of tasks. For example, the L_2 error between the model output and the true output, averaged over the input data, is a common term in the loss function. In addition, other terms may be added to regularize the optimization (e.g., the L_1 or L_2 norm of the parameters $\boldsymbol{\theta}$ to promote parsimony and prevent overfitting). Thus, the loss function typically balances multiple competing objectives, such as model performance and model complexity. The loss function may also incorporate terms used to promote a specific behavior across different sub-networks in a neural network architecture. Importantly, the loss function will provide valuable information used to approximate gradients required to optimize the parameters.

2.4.1 Embedding physics

Most of the physics-informed architectures described above involve custom loss functions to promote the efficient training of accurate models. It is also possible to incorporate physical priors, such as sparsity, by adding L_1 or L_0 regularizing loss terms on the parameters in $\boldsymbol{\theta}$. In fact, parsimony has been a central theme in physical modeling for century, where it is believed that balancing model complexity with descriptive capability is essential in developing models that generalize. The sparse identification of nonlinear dynamics algorithm [18] learns dynamical systems models with as few terms from a library of candidate terms as are needed to describe the training data. There are several formulations involving different loss terms and optimization algorithms that promote additional physical notions, such as stability [99] and energy conservation [100]. Stability promoting loss functions based on notions of Lyapunov stability have also been incorporated into autoencoders, with impressive results on fluid systems [101].

2.4.2 Examples in fluid mechanics

Sparse nonlinear modeling has been used extensively in fluid mechanics, adding sparsity-promoting loss terms to learn parsimonious models that prevent overfitting and generalize to new scenarios. SINDy has been used to generate reduced-order models for how dominant coherent structures evolve in a flow for a range of configurations [100,102–105]. These models have also been extended to develop compact closure models [87–89]. Recently, the physical notion of *boundedness* of solutions, which is a fundamental concept in reduced-order models of fluids [106], has been incorporated into the SINDy modeling framework as a novel loss function. Other physical loss functions may be added, such as adding the divergence of a flow field as a loss term to promote solutions that are incompressible [107].

2.5 The optimization algorithm

Ultimately, machine learning models are trained using optimization algorithms to find the parameters $\boldsymbol{\theta}$ that best fit the training data. Typically, these optimization problems are both high-dimensional and non-convex, leading to extremely challenging optimization landscapes with many local minima. While there are powerful and generic techniques for convex optimization problems [108,109], there are few generic guarantees for convergence or global optimality in non-convex optimization. Modern deep neural networks have particularly high-dimensional parameters $\boldsymbol{\theta}$ and require large training data sets, which necessitate stochastic gradient descent algorithms. In a sense, the optimization algorithm is the engine powering machine learning, and as such, it is often abstracted from the decision process. However, developing advanced optimization algorithms is the focus of intense research efforts. It is also often necessary to explicitly consider the optimization algorithm when designing a new architecture or incorporating a novel loss term.

2.5.1 Embedding physics

There are several ways that the optimization algorithm may be customized or modified to incorporate prior physical

knowledge. One approach is to explicitly add constraints to the optimization, for example that certain coefficients must be non-negative, or that other coefficients must satisfy a specified algebraic relationship with each other. Depending on the given machine learning architecture, it may be possible to enforce energy conservation [100] or stability constraints [99] in this way. Another approach involves employing custom optimization algorithms required to minimize the physically motivated loss functions above, which are often non-convex. In this way, the line between loss function and optimization algorithm are often blurred, as they are typically tightly coupled. For example, promoting sparsity with the L_0 norm is non-convex, and several relaxed optimization formulations have been developed to approximately solve this problem. The sparse relaxed regularized regression (SR3) optimization framework [68] has been developed specifically to handle challenging non-convex loss terms that arise in physically motivated problems.

2.5.2 Examples in fluid mechanics

Loiseau [100] showed that it is possible to enforce energy conservation for incompressible fluid flows directly by imposing skew-symmetry constraints on the quadratic terms of a sparse generalized linear regression (i.e. SINDy) model. These constraints manifest as equality constraints on the sparse coefficients $\boldsymbol{\theta}$ of the SINDy model. Because the standard SINDy optimization procedure is based on a sequentially thresholded least-squares procedure, it is possible to enforce these equality constraints at every stage of the regression, using the Karush–Kuhn–Tucker (KKT) conditions. The SR3 optimization package [68] was developed to generalize and extend these constrained optimization problems to more challenging constraints, and to more generic optimization problems. This is only one of many examples of custom optimization algorithms being developed to train machine learning models with novel loss functions or architectures.

3 Parting thoughts

This brief paper has attempted to provide a high level overview of the various stages of machine learning, how physics can be incorporated at each stage, and how these techniques are being applied today in fluid mechanics. Machine learning for physical systems requires careful consideration in each of these steps, as every stage provides an opportunity to incorporate prior knowledge about the physics. A working definition of physics is the part of a model that generalizes, and this is one of the central goals of machine learning models for physical systems. It is also important to note that machine learning is fundamentally a collaborative effort, as it is nearly impossible to master every stage of this process.

The nature of this topic is mercurial, as new innovations are being introduced every day that improve our capabilities and challenge our previous assumptions. Much of this work has deliberately oversimplified the process of machine learning and the field of fluid mechanics. Machine learning is largely concerned with fitting functions from data, and so it is important to pick the right functions to fit. The inputs to the function are the variables and parameters that we have access to or control over, and the outputs are quantities of interest that we would like to accurately and efficiently approximate in the future. It is a fruitful exercise to revisit classically important problems where progress was limited by our ability to represent complex functions. For example, Ling et al. [30] had great success revisiting the classical Reynolds stress models of Pope [110] with powerful modern techniques. More fundamentally, machine learning is about asking and answering questions with data. We can't forget why we are asking these questions in the first place: because we are curious, and there is value in knowing the answer.

Acknowledgements SLB acknowledges many valuable discussions and perspectives gained from collaborators and coauthors Petros Koumoutsakos, J. Nathan Kutz, Jean-Christophe Loiseau, and Bernd Noack.

Disclaimer Any omission or oversight was the result of either ignorance, forgetfulness, hastiness, or lack of imagination on my part. These notes are not meant to be exhaustive, but rather to provide a few concrete examples from the literature to guide researchers getting started in this field. This field is growing at an incredible rate, and these examples provide a tiny glimpse into a much larger effort. I have tried to sample from what I consider some of the most relevant and accessible literature. However, a disproportionate number of references are to work by my close collaborators, as this is the work I am most familiar with. If I have missed any important references or connections, or mis-characterized any works cited here, please let me know and I'll try to incorporate corrections in future versions of these notes.

References

1. Taira, K., Brunton, S.L., Dawson, S., et al.: Modal analysis of fluid flows: An overview. AIAA J. **55**, 4013–4041 (2017)
2. Rabault, J., Kuchta, M., Jensen, A., et al.: Artificial neural networks trained through deep reinforcement learning discover

control strategies for active flow control. J. Fluid Mech. **865**, 281–302 (2019)
3. Ren, F., Hu, H.B., Tang, H.: Active flow control using machine learning: A brief review. J. Hydrodyn. **32**, 247–253 (2020)
4. Zhou, Y., Fan, D., Zhang, B., et al.: Artificial intelligence control of a turbulent jet. J. Fluid Mech. **897**, A27 (2020)
5. Brunton, S.L., Noack, B.R., Koumoutsakos, P.: Machine learning for fluid mechanics. Annu. Rev. Fluid Mech. **52**, 477–508 (2020)
6. Du, M., Liu, N., Hu, X.: Techniques for interpretable machine learning. Commun. ACM **63**, 68–77 (2019)
7. Molnar, C.: Interpretable machine learning. Lulu. com (2020)
8. Duraisamy, K., Iaccarino, G., Xiao, H.: Turbulence modeling in the age of data. Annu. Rev. Fluid Mech. **51**, 357–377 (2019)
9. Brenner, M., Eldredge, J., Freund, J.: Perspective on machine learning for advancing fluid mechanics. Phys. Rev. Fluids **4**, 100501 (2019)
10. Brenner, M.P., Koumoutsakos, P.: Machine learning and physical review fluids: An editorial perspective. Phys. Rev. Fluids **6**, 070001 (2021)
11. Brunton, S.L., Kutz, J.N.: Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Cambridge University Press, Cambridge (2019)
12. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, vol. 1. MIT Press, Cambridge (1998)
13. Mnih, V., Kavukcuoglu, K., Silver, D., et al.: Human-level control through deep reinforcement learning. Nature **518**, 529 (2015)
14. Silver, D., Schrittwieser, J., Simonyan, K., et al.: Mastering the game of go without human knowledge. Nature **550**, 354–359 (2017)
15. Kaiser, E., Kutz, J.N., Brunton, S.L.: Discovering conservation laws from data for control. In: 2018 IEEE Conference on Decision and Control (CDC), pp. 6415–6421. IEEE (2018)
16. Schmidt, M., Lipson, H.: Distilling free-form natural laws from experimental data. Science **324**, 81–85 (2009)
17. Schmid, P.J.: Dynamic mode decomposition of numerical and experimental data. J. Fluid Mech. **656**, 5–28 (2010)
18. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **113**, 3932–3937 (2016)
19. Pathak, J., Lu, Z., Hunt, B.R., et al.: Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. Chaos: An Interdisciplinary. J. Nonlinear Sci. **27**, 121102 (2017)
20. Vlachas, P.R., Byeon, W., Wan, Z.Y., et al.: Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. Proc. R. Soc. A **474**, 20170844 (2018)
21. Lusch, B., Kutz, J.N., Brunton, S.L.: Deep learning for universal linear embeddings of nonlinear dynamics. Nat. Commun. **9**, 4950 (2018)
22. Wehmeyer, C., Noé, F.: Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. J. Chem. Phys. **148**, 1–9 (2018)
23. Mardt, A., Pasquali, L., Wu, H., et al.: VAMPnets: Deep learning of molecular kinetics. Nat. Commun. **9**, 5 (2018)
24. Takeishi, N., Kawahara, Y., Yairi, T.: Learning koopman invariant subspaces for dynamic mode decomposition. In: Advances in Neural Information Processing Systems, pp. 1130–1140 (2017)
25. Li, Q., Dietrich, F., Bollt, E.M., et al.: Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator. Chaos: An Interdisciplinary. J. Nonlinear Sci. **27**, 103111 (2017)
26. Yeung, E., Kundu, S., Hodas, N.: Learning deep neural network representations for koopman operators of nonlinear dynamical systems. arXiv:1708.06850 (2017)
27. Otto, S.E., Rowley, C.W.: Linearly-recurrent autoencoder networks for learning dynamics. SIAM J. Appl. Dyn. Syst. **18**, 558–593 (2019)
28. Champion, K., Lusch, B., Kutz, J.N., et al.: Data-driven discovery of coordinates and governing equations. Proc. Natl. Acad. Sci. **116**, 22445–22451 (2019)
29. Ahmed, S.E., Pawar, S., San, O., et al.: On closures for reduced order models — a spectrum of first-principle to machine-learned avenues. arXiv:2106.14954 (2021)
30. Ling, J., Kurzawski, A., Templeton, J.: Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. J. Fluid Mech. **807**, 155–166 (2016)
31. Kutz, J.N.: Deep learning in fluid dynamics. J. Fluid Mech. **814**, 1–4 (2017)
32. Maulik, R., San, O., Rasheed, A., et al.: Subgrid modelling for two-dimensional turbulence using neural networks. J. Fluid Mech. **858**, 122–144 (2019)
33. Novati, G., de Laroussilhe, H.L., Koumoutsakos, P.: Automating turbulence modelling by multi-agent reinforcement learning. Nat. Mach. Intell. **3**, 87–96 (2021)
34. Wang, J.X., Wu, J.L., Xiao, H.: Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. Phys. Rev. Fluids **2**, 034603 (2017)
35. Zhu, L., Zhang, W., Kou, J., et al.: Machine learning methods for turbulence modeling in subsonic flows around airfoils. Phys. Fluids **31**, 015105 (2019)
36. Zhu, L., Zhang, W., Sun, X., et al.: Turbulence closure for high Reynolds number airfoil flows by deep neural networks. Aerosp. Sci. Technol. **110**, 106452 (2021)
37. Bar-Sinai, Y., Hoyer, S., Hickey, J., et al.: Learning data-driven discretizations for partial differential equations. Proc. Natl. Acad. Sci. **116**, 15344–15349 (2019)
38. Thaler, S., Paehler, L., Adams, N.A.: Sparse identification of truncation errors. J. Comput. Phys. **397**, 108851 (2019)
39. Stevens, B., Colonius, T.: Enhancement of shock-capturing methods via machine learning. Theoret. Comput. Fluid Dyn. **34**, 483–496 (2020)
40. Kochkov, D., Smith, J.A., Alieva, A., et al.: Machine learning accelerated computational fluid dynamics. arXiv:2102.01010 (2021)
41. Erichson, N.B., Mathelin, L., Yao, Z., et al.: Shallow neural networks for fluid flow reconstruction with limited sensors. Proc. R. Soc. A **476**(2238), 20200097 (2020)
42. Fukami, K., Fukagata, K., Taira, K.: Super-resolution reconstruction of turbulent flows with machine learning. J. Fluid Mech. **870**, 106–120 (2019)
43. Taira, K., Hemati, M.S., Brunton, S.L., et al.: Modal analysis of fluid flows: Applications and outlook. AIAA J. **58**(3), 998–1022 (2020)
44. Scherl, I., Strom, B., Shang, J.K., et al.: Robust principal component analysis for particle image velocimetry. Phys. Rev. Fluids **5**, 054401 (2020)
45. Nair, A.G., Taira, K.: Network-theoretic approach to sparsified discrete vortex dynamics. J. Fluid Mech. **768**, 549–571 (2015)
46. Kaiser, E., Noack, B.R., Cordier, L., et al.: Cluster-based reduced-order modelling of a mixing layer. J. Fluid Mech. **754**, 365–414 (2014)
47. Fernex, D., Noack, B.R., Semaan, R.: Cluster-based network modeling–from snapshots to complex dynamical systems. Science. Advances **7**, eabf5006 (2021)
48. Maceda, G.Y.C., Li, Y., Lusseyran, F., et al.: Stabilization of the fluidic pinball with gradient-enriched machine learning control. J. Fluid Mech. **917**, 45 (2021)

49. Fan, D., Yang, L., Wang, Z., et al.: Reinforcement learning for bluff body active flow control in experiments and simulations. Proc. Natl. Acad. Sci. **117**, 26091–26098 (2020)
50. Verma, S., Novati, G., Koumoutsakos, P.: Efficient collective swimming by harnessing vortices through deep reinforcement learning. Proc. Natl. Acad. Sci. **115**, 5849–5854 (2018)
51. Fan, D., Jodin, G., Consi, T., et al.: A robotic intelligent towing tank for learning complex fluid–structure dynamics. Sci. Robot. **4**, 36 (2019)
52. Kou, J., Zhang, W.: Data-driven modeling for unsteady aerodynamics and aeroelasticity. Prog. Aerosp. Sci. **125**, 100725 (2021)
53. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp. 1097–1105 (2012)
54. Deng, J., Dong, W., Socher, R., et al.: Imagenet: A largescale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255. IEEE (2009)
55. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
56. Meng, X., Karniadakis, G.E.: A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. J. Comput. Phys. **401**, 109020 (2020)
57. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**, 359–366 (1989)
58. Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural Netw. **4**, 251–257 (1991)
59. Wu, X., Kumar, V., Quinlan, J.R., et al.: Top 10 algorithms in data mining. Knowl. Inf. Syst. **14**, 1–37 (2008)
60. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
61. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, Cambridge (2002)
62. Blanchard, A., Sapsis, T.: Bayesian optimization with output-weighted optimal sampling. J. Comput. Phys. **425**, 109901 (2021)
63. Bongard, J., Lipson, H.: Automated reverse engineering of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **104**, 9943–9948 (2007)
64. Cranmer, M.D., Xu, R., Battaglia, P., et al.: Learning symbolic physics with graph networks. arXiv:1909.05862 (2019)
65. Cranmer, M., Sanchez-Gonzalez, A., Battaglia, P., et al.: Discovering symbolic models from deep learning with inductive biases. arXiv:2006.11287 (2020)
66. Kutz, J.N., Brunton, S.L., Brunton, B.W., et al.: Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems. SIAM, Bangkok (2016)
67. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B (Methodological) **58**, 267–288 (1996)
68. Zheng, P., Askham, T., Brunton, S.L., et al.: Sparse relaxed regularized regression: SR3. IEEE Access **7**, 1404–1423 (2019)
69. Pathak, J., Hunt, B., Girvan, M., et al.: Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. Phys. Rev. Lett. **120**, 024102 (2018)
70. Li, K., Kou, J., Zhang, W.: Deep neural network for unsteady aerodynamic and aeroelastic modeling across multiple mach numbers. Nonlinear Dyn. **96**, 2157–2177 (2019)
71. Thomas, N., Smidt, T., Kearnes, S., et al.: Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. arXiv:1802.08219 (2018)
72. Miller, B.K., Geiger, M., Smidt, T.E., et al.: Relevance of rotationally equivariant convolutions for predicting molecular properties. arXiv:2008.08461 (2020)
73. Wang, R., Walters, R., Yu, R.: Incorporating symmetry into deep dynamics models for improved generalization. arXiv:2002.03061 (2020)
74. Batzner, S., Smidt, T.E., Sun, L., et al.: Se (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. arXiv:2101.03164 (2021)
75. Greydanus, S., Dzamba, M., Yosinski, J.: Hamiltonian neural networks. Adv. Neural Inf. Process. Syst. **32**, 15379–15389 (2019)
76. Finzi, M., Wang, K.A., Wilson, A.G.: Simplifying Hamiltonian and Lagrangian neural networks via explicit constraints. Adv. Neural Inf. Process. Syst. **33**, 1–10 (2020)
77. Cranmer, M., Greydanus, S., Hoyer, S., et al.: Lagrangian neural networks. arXiv:2003.04630 (2020)
78. Zhong, Y.D., Leonard, N.: Unsupervised learning of Lagrangian dynamics from images for prediction and control. Adv. Neural Inf. Process. Syst. **33**, 1–12 (2020)
79. Raissi, M., Perdikaris, P., Karniadakis, G.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)
80. Pang, G., Lu, L., Karniadakis, G.E.: fpinns: Fractional physics-informed neural networks. SIAM J. Sci. Comput. **41**, A2603–A2626 (2019)
81. Yang, L., Zhang, D., Karniadakis, G.E.: Physics-informed generative adversarial networks for stochastic differential equations. SIAM J. Sci. Comput. **42**, A292–A317 (2020)
82. Mao, Z., Jagtap, A.D., Karniadakis, G.E.: Physics-informed neural networks for high-speed flows. Comput. Methods Appl. Mech. Eng. **360**, 112789 (2020)
83. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., et al.: Physics-informed machine learning. Nat. Rev. Phys. **3**, 422–440 (2021)
84. Battaglia, P.W., Hamrick, J.B., Bapst, V., et al.: Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261 (2018)
85. Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., et al.: Learning to simulate complex physics with graph networks. In: International Conference on Machine Learning, pp. 8459–8468. PMLR (2020)
86. Lu, L., Jin, P., Pang, G., et al.: Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. Nat. Mach. Intell. **3**, 218–229 (2021)
87. Beetham, S., Capecelatro, J.: Formulating turbulence closures using sparse regression with embedded form invariance. Phys. Rev. Fluids **5**, 084611 (2020)
88. Beetham, S., Fox, R.O., Capecelatro, J.: Sparse identification of multiphase turbulence closures for coupled fluid–particle flows. J. Fluid Mech. **914**, A11 (2021)
89. Schmelzer, M., Dwight, R.P., Cinnella, P.: Discovery of algebraic Reynolds-stress models using sparse symbolic regression. Flow Turbul. Combust. **104**(2), 579–603 (2020)
90. Kou, J., Zhang, W.: A hybrid reduced-order framework for complex aeroelastic simulations. Aerosp. Sci. Technol. **84**, 880–894 (2019)
91. Raissi, M., Yazdani, A., Karniadakis, G.E.: Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. Science **367**, 1026–1030 (2020)
92. Zhao, X., Du, L., Peng, X., et al.: Research on refined reconstruction method of airfoil pressure based on compressed sensing. Theoret. Appl. Mech. Lett. **11**, 100223 (2021)
93. Li, Z., Kovachki, N., Azizzadenesheli, K., et al.: Fourier neural operator for parametric partial differential equations. arXiv:2010.08895 (2020)
94. Frezat, H., Balarac, G., Le Sommer, J., et al.: Physical invariance in neural networks for subgrid-scale scalar flux modeling. Phys. Rev. Fluids **6**(2), 024607 (2021)
95. Lee, K., Carlberg, K.T.: Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. J. Comput. Phys. **404**, 108973 (2020)

96. Noack, B.R., Afanasiev, K., Morzynski, M., et al.: A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. J. Fluid Mech. **497**, 335–363 (2003)
97. Benner, P., Gugercin, S., Willcox, K.: A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev. **57**, 483–531 (2015)
98. Rowley, C.W., Dawson, S.T.: Model reduction for flow analysis and control. Annu. Rev. Fluid Mech. **49**, 387–417 (2017)
99. Kaptanoglu, A.A., Callaham, J.L., Hansen, C.J., et al.: Promoting global stability in data-driven models of quadratic nonlinear dynamics. arXiv:2105.01843 (2021)
100. Loiseau, J.C., Brunton, S.L.: Constrained sparse Galerkin regression. J. Fluid Mech. **838**, 42–67 (2018)
101. Erichson, N.B., Muehlebach, M., Mahoney, M.W.: Physics-informed autoencoders for lyapunov-stable fluid flow prediction. arXiv:1905.10866 (2019)
102. Loiseau, J.C., Noack, B.R., Brunton, S.L.: Sparse reduced-order modeling: Sensor-based dynamics to full-state estimation. J. Fluid Mech. **844**, 459–490 (2018)
103. Loiseau, J.C.: Data-driven modeling of the chaotic thermal convection in an annular thermosyphon. Theoret. Comput. Fluid Dyn. **34**, 339–365 (2020)
104. Deng, N., Noack, B.R., Morzyński, M., et al.: Low-order model for successive bifurcations of the fluidic pinball. J. Fluid Mech. **884**, A37 (2020)
105. Deng, N., Noack, B.R., Morzyński, M., et al.: Galerkin force model for transient and post-transient dynamics of the fluidic pinball. J. Fluid Mech. **918**, A4 (2021)
106. Schlegel, M., Noack, B.R.: On long-term boundedness of Galerkin models. J. Fluid Mech. **765**, 325–352 (2015)
107. Wang, R., Kashinath, K., Mustafa, M., et al.: 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)
108. Grant, M., Boyd, S., Ye, Y.: Cvx: Matlab software for disciplined convex programming (2008)
109. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2009)
110. Pope, S.: A more general effective-viscosity hypothesis. J. Fluid Mech. **72**, 331–340 (1975)