Perspective

Physics-informed and data-driven discovery of governing equations for complex phenomena in heterogeneous media

Muhammad Sahimi
Mork Family Department of Chemical Engineering and Materials Science, University of Southern California, Los Angeles, California 90089-1211, USA

(Received 9 February 2023; revised 7 September 2023; published 4 April 2024)

Rapid evolution of sensor technology, advances in instrumentation, and progress in devising data-acquisition software and hardware are providing vast amounts of data for various complex phenomena that occur in heterogeneous media, ranging from those in atmospheric environment, to large-scale porous formations, and biological systems. The tremendous increase in the speed of scientific computing has also made it possible to emulate diverse multiscale and multiphysics phenomena that contain elements of stochasticity or heterogeneity, and to generate large volumes of numerical data for them. Thus, given a heterogeneous system with annealed or quenched disorder in which a complex phenomenon occurs, how should one analyze and model the system and phenomenon, explain the data, and make predictions for length and time scales much larger than those over which the data were collected? We divide such systems into three distinct classes. (i) Those for which the governing equations for the physical phenomena of interest, as well as data, are known, but solving the equations over large length scales and long times is very difficult. (ii) Those for which data are available, but the governing equations are only partially known, in the sense that they either contain various coefficients that must be evaluated based on the data, or that the number of degrees of freedom of the system is so large that deriving the complete equations is very difficult, if not impossible, as a result of which one must develop the governing equations with reduced dimensionality. (iii) In the third class are systems for which large amounts of data are available, but the governing equations for the phenomena of interest are not known. Several classes of physics-informed and data-driven approaches for analyzing and modeling of the three classes of systems have been emerging, which are based on machine learning, symbolic regression, the Koopman operator, the Mori-Zwanzig projection operator formulation, sparse identification of nonlinear dynamics, data assimilation combined with a neural network, and stochastic optimization and analysis. This perspective describes such methods and the latest developments in this highly important and rapidly expanding area and discusses possible future directions.

DOI: 10.1103/PhysRevE.109.041001

I. INTRODUCTION

A wide variety of systems of scientific, industrial, and societal importance represent heterogeneous, and multiphase and multiscale media. Examples vary anywhere from large-scale porous formations, to composite materials, biological systems, and Earth's atmosphere. Many complex phenomena also occur in such systems, including fluid flow, transport, reaction, and deformation. Given the extreme importance of such systems to human and societal progress, the goal for decades has been developing models that describe not only the multiscale and multiphase systems themselves, but also the phenomena that occur there.

Consider, as an example, the problem of air pollution in large urban areas. Chemical oxidants, especially ozone, are major products of photochemical oxidation (reactions that are influenced by Sun) of primary pollutants emitted from various sources in the tropospheric layer [1]. Although the presence of ozone in the stratospheric layer is responsible for continuation of life on Earth, its presence in the troposphere is dangerous to humans' health and damaging to national and international economies [2]. Effective control of the pollutants requires accurate and comprehensive knowledge of the rates of emission and transport of the reactants that are present in the atmosphere, and the chemical reactions that they participate in. In particular, since in the presence of nitrogen oxides, NO_x, ozone production increases very significantly [1], which is the main cause of the formation of photochemical smog, detailed information on its concentration is needed for its control.

Such data are continuously collected by a large number of sensors in large urban areas around the world, and have been becoming available. To analyze and understand the huge volume of the data that are being continuously collected, modeling of such phenomena has been pursued for decades. Large urban areas are, however, highly complex media. Consider, for example, the Greater Tehran area, Iran's Capital, which begins on the tall Alborz mountains in the north, and ends in the desert in the south, or the Greater Los Angeles area that is sandwiched between San Bernardino, San Gabriel, and San Fernando mountains and the Pacific Ocean. Clearly, the terrains and topography of such large urban areas are highly rough and complex. Any modeling of atmospheric pollution over the two areas must take into acount not only the effect of the large rough terrains—about 1300 and 87 000 km^2 for, respectively, the Greater Tehran and Los Angeles areas—and their rough topography, but also the dynamic changes that

occur there continuously on hourly, daily, monthly, and seasonal bases, as the two areas represent multiscale systems, not only in space, but also in time, which span at least 10–15 orders of magnitude. As a result, numerical simulation of such complex phenomena, even if the governing equations are known, is extremely difficult, as it involves turbulent flow, reactions with highly nonlinear kinetics, a huge number of reactants—typically five dozens or more—and the reaction products—over one hundred -the presence or absence of a wind velocity field, boundary conditions that vary dynamically, and many other complicating factors [1,3].

The availability of vast amount of data is not limited to the problem of atmospheric pollution. It is estimated that, over the next decade, hundreds of billions of sensors that include airborne, seaborne and satellite remote sensing will be collecting vast amounts of data for many phenomena, such as vegetation and plantation, and the characteristics of draught-stricken areas, an increasingly important problem worldwide. The same is also true of large geomedia and such complex problems as seismology, fracture propagation, and earthquakes. Analysis of such data, particularly those for which the signal-to-noise ratio is low—usually referred to as noisy data—understanding the subtle insights that they may provide, and incorporating them into accurate physical models is a Herculean task, requiring a paradigm shift.

Such a paradigm shift has slowly begun to emerge, with two classes of approaches are currently being developed. One class exploits deep-learning (DL), and more generally machine-learning (ML), algorithms to address the problem. The approach has been motivated by the fact that in many cases, the ML algorithms [4,5] are capable of extracting important features from vast amounts of data that are characterized by spatial and temporal coverage; see for example, Reichstein *et al.* [6]. In some cases, the governing equations for the complex phenomena for which the data have been collected may be known, which are then incorporated into a ML approach to develop predictive tools for studying the phenomena over spatial and time scales well beyond those over which the existing data have been collected. Two representative examples of such approaches are the work of Kamrava *et al.* [7] for modeling fluid flow in porous materials, and that of Alber *et al.* [8] for modeling of biophysical and biomedical systems. In other cases, the governing equations may be known, but they contain transport and other types of coefficients that depend on the morphology of the systems in which a complex phenomena occur, or require constitutive relationships without which the governing equations would not be very useful, unless one resorts to pure empiricism. As a result, the constitutive relationships and/or the coefficients that the governing equations contain must be *discovered*.

The second class of approaches is intended for the systems in which the governing equations for physical phenomena occurring in them and, hence, for the associated data, are not known. Thus, one attempts to discover the equations using the large amount of data currently available. The lack of governing equations is particularly true for those phenomena that involve multiscale heterogeneity in the form of some sort of stochasticity. The discovery of such equations has dominated physical sciences and engineering for the past several decades, as they provide predictions for systems' behavior.

The classical approach has been based on the fundamental conservation laws, namely, the equations that describe mass, momentum and energy conservation. If a system is heterogeneous, then the microscale conservation laws are averaged over an ensemble of its possible realizations to derive the macroscale equations. This is, however, valid only if there is a well-defined representative elementary volume (REV) or scale, i.e., the volume or length scale over which the heterogeneous system can be considered as macroscopically homogeneous, so that it is stationary over length scales larger than the REV.

But, what if the REV does not exist, or is larger than the size of the system, in which case the system is nonstationary, i.e., the probability distribution functions (PDFs) of its properties vary spatially from region to region? Examination of many important systems indicates that nonstationarity is more like the rule, rather than the exception. A good example is natural porous media at large (regional) length scales. It is known [9] that the physical properties of such media, such as their permeability and elastic constants approximately follow nonstationary stochastic functions [10]. Thus, the question is, what are the governing equations for flow, transport, and deformation processes in such media?

Other obvious examples are biological, and nano- and neuroscience systems for which first-principle calculations are currently very difficult, if not impossible, to carry out, whereas data for them are becoming abundant and, in many cases, with exceptional quality. In addition, the tremendous increase in the computational power is making it possible to emulate the behavior of diverse and complex systems that are high-dimensional, multiscale, and stochastic. The question, then, is, how can we discover the governing equations that not only honor and better explain the data, but also provide predictions for the future, or over much larger length and timescales than those over which the data were collected? It should be clear that the ability to discover the governing equations based directly on the data is of paramount importance in many modern scientific and engineering problems.

This perspective describes the emerging field of physics-informed and data-driven (PIDD) modeling of multiscale, multiphysics systems and phenomena and, in particular, the approaches for discovering the governing equations for given sets of data that represent the characteristics of complex phenomenon in heterogeneous media. We describe the emerging approaches, discuss their strengths and shortcomings, and point out possible future directions in this rapidly developing and highly significant research area.

II. THREE TYPES OF SYSTEMS

This perspective is *not* about all the various computational approaches for modeling of heterogeneous media. Instead, it tries to address two fundamental questions: (a) Given a set of data for a complex phenomenon in a heterogeneous system, what is the best PIDD approach for modeling the phenomenon? (b) If the governing equations for the phenomenon are unknown, then how can one discover them through a PIDD method? In general, the success of any PIDD approach for addressing the two questions depends on the amount of available data, on the one hand, and the structure and complexity

of the system itself, on the other hand. Thus, let us divide such systems of interests into three classes:

(i) Type I, which consists of those systems for which the governing equations for the physical phenomena of interest, together with data for the phenomena, are known, but simulating them in the heterogeneous system over large length and long timescales is very difficult, if not impossible. For example, Darcy's law together with the Stokes' equation describe slow flow of Newtonian fluids in microscopically disordered, but macroscopically homogeneous porous media, while the convective-diffusion equation describes transport of a solute and mass transfer in the same media [9], but solving such equations in large-scale heterogeneous porous media over long timescales is very difficult. In this case, the goal is to develop a PIDD approach to simulate fluid flow or transport process at large length scales and long times.

(ii) Type II represents systems for which data for complex phenomena in the systems are available, and the physics of the phenomena of interest is also partially known. For example, any fluid flow and transport problem in porous media is governed by the equations that describe mass and momentum conservation, but they contain coefficients, such as the permeability, diffusivity, and the dispersion coefficients, which must be evaluated based on the available data. Another example is when the available data are for a system that exhibits multiscale features, in which case reconstructing the complete governing equations is a very difficult task, often involving prohibitive computations. In such a case one resorts to a PIDD dimensionality reduction technique, i.e., discovering an ordinary or partial differential equation with the lowest order that describes the data.

(iii) Type III systems represent the opposite to Type I, i.e., those for which large amounts of data for some complex phenomena are available, but the governing equations for the phenomena at the macroscale are not known. Thus, the goal is developing a PIDD algorithm for discovering the governing equations for the phenomena of interest, and understanding and explaining the data. As we describe below, one approach to address this problem may be based on dimensionality reduction, but other computational methods have also been emerging.

In the rest of this perspective we describe the PIDD approaches for each of the three classes of systems.

III. TYPE-I SYSTEMS

We begin our discussions by describing the emerging PIDD approaches for Type-I systems.

A. Data assimilation

Data assimilation is a well-established concept that has been utilized in the investigations of the atmospheric and geological sciences to make concrete predictions for weather, oceans, climate, and ecosystems, as well as geomedia. Since data assimilation techniques improve forecasting, or help developing a more accurate model that provides us with deeper understanding of such complex systems, they play an important role in the studies of climate change and pollution of environment, as well as geological systems.

Data assimilation combines observational data with the dynamical principles, or the equations or models that govern a system of interest, to obtain an estimate of its state that is more accurate than what is obtained by using only the data or the physical model alone. Thus, in essence, data assimilation is suitable for the first type of systems described in Sec. III, i.e., those for which some reasonable amounts of data are available, and the physics of the phenomena of interest is also known. Both the data and the models have errors, however. As discussed by Zhang and Moore [11], the errors in the data are of random, systematic, and representativeness types. Models also produce errors because, often, they are simplified, or are incomplete to begin with, to make the computations affordable, which in turn generates error. We do not intend to review in detail data assimilation methods, as they are well known, but only describe them briefly, since in the next subsection we discuss how data assimilation methods can be combined with a machine-learning algorithm to not only improve forecasting, but also reduce the computational burden significantly.

There are at least four approaches to data assimilation, which are the Cressman method, the optimal interpolation method, three- or four-dimensional variational analysis, and the Kalman filter. They all represent least-squares methods, with the final estimate selected in such a way as to minimize the uncertainty. In all four approaches, the set of data representing a system's state is denoted by $\mathbf{x}$. The actual or true state $\mathbf{x}_t$ is different from the best possible representation $\mathbf{x}_b$, produced by physical models and referred to as the background state. To analyze the system and the data, an observation vector $\mathbf{y}$ is compared with the state vector.

In the Cressman method [12], which belongs to a class of methods called objective analysis, one assumes that the model state is univariate and is represented by values of the variable at discrete grid points. Suppose that a previous best estimate of the model state is an n-dimensional vector, $\mathbf{x}_b = [x_b(1), \cdots, x_b(n)]^{\mathrm{T}}$, while the observations are represented by a n-dimensional vector, $\mathbf{y} = [y(1), \cdots, y(n)]^{\mathrm{T}}$. The Cressman method provides an updated model, $\mathbf{x}_u = [x_u(1), \cdots, x_u(n)]^{\mathrm{T}}$, by the following equation:

$$\mathbf{x}_u(j) = \mathbf{x}_b(j) + \frac{\sum_{i=1}^{n} \omega_{ij}[\mathbf{y}(i) - \mathbf{x}_b(i)]}{\sum_{i=1}^{n} \omega_{ij}}, \tag{1}$$

with $\omega_{ij} = \max[0, (R^2 - d_{ij}^2)/(R^2 + d_{ij}^2)]$ and $d_{ij} = |i - j|$. Note that $\omega_{ij} = 1$, if $i = j$, and $\omega_{ij} = 0$, if $d_{ij} > R$. R, which is a control parameter defined by the user, is referred to as the influence parameter.

In the optimal interpolation method one combines the observation vector $\mathbf{y}$ with p entries with the background vector $\mathbf{x}_b$ wth n entries, with $n \geqslant p$. Because there are usually fewer observations than variables in the background model, the only correct way of making the comparison is to use an observation operator h from model n-dimensional state space to p-dimensional observation space, which is a $p \times n$ matrix $\mathbf{H}$ such that $h(\mathbf{x}_b) = (h_1, \cdots, h_p)^{\mathrm{T}} = \mathbf{H}\mathbf{x}_b$, with $h_i = \sum_{j=1}^{n} H_{ij} x_b(j)$.

Suppose that $\mathbf{B}$ with a size $n \times n$ and $\mathbf{R}$ with a size $p \times p$ are, respectively, the covariance matrices of the background error $\mathbf{x}_b - \mathbf{x}_t$, and observation error $\mathbf{y} - h(\mathbf{x}_b)$. The two errors are assumed to be uncorrelated. The n-dimensional analysis,

or updated, vector $\mathbf{x}_u$ is defined by, $\mathbf{x}_u = \mathbf{x}_b + \mathbf{w}[\mathbf{y} - h(\mathbf{x}_b)]$, where $\mathbf{w}$ is an $n \times p$ matrix that is selected such that the variance of $\mathbf{x}_u - \mathbf{x}_t$ is minimized. It can be shown that, $\mathbf{w} = \mathbf{B}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}$.

In the three-dimensional variational analysis, a cost function $\sigma^2(\mathbf{x})$ is defined by

$$\sigma^2(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_b)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + [\mathbf{y} - h(\mathbf{x})]^{\mathrm{T}}\mathbf{R}^{-1}[\mathbf{y} - h(\mathbf{x})] \equiv \sigma_b^2(\mathbf{x}) + \sigma_o^2(\mathbf{x}), \tag{2}$$

with σ_b^2 and σ_o^2 being the background and observation cost functions. It has been proven that if we write $\mathbf{x} = \mathbf{x}_u = \mathbf{x}_b + \mathbf{w}[\mathbf{y} - h(\mathbf{x})_b]$, then the cost function attains its global minimum.

Generalization of the method to four-dimensional variational assimilation is straightforward. The observations are distributed among $(N + 1)$ times in the interval of interest. The cost function is defined by

$$\sigma^2(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_b)^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + \sum_{i=0}^{N}[\mathbf{y}_i - H_i(\mathbf{x}_i)]^{\mathrm{T}}R_i^{-1}[\mathbf{y}_i - H_i(\mathbf{x}_i)], \tag{3}$$

and, therefore, the data assimilation problem with a globally minimum variance is reduced to computing the vector $\mathbf{x}_u$ such that $\sigma^2(\mathbf{x})$ attains its minimal at $\mathbf{x} = \mathbf{x}_u$.

The Kalman filter [13], also known as linear quadratic estimation, has been used to continuously update the parameters of models of dynamical systems for assimilating data. The filter is optimal only under the assumption that the system is linear and the measurement and process noises follow Gaussian distributions. The algorithm, a recursive one, consists of two steps. In the prediction step, the filter generates estimates of the current state variables $\mathbf{x}$ together with their uncertainties. After the data for the next measurement, which may contain some error, become available, step two commences in which the estimates are updated using a weighted average, with more weight given to those with greater certainty (smaller errors). The algorithm operates in real time, using only the present input measurements, and the state calculated previously and its uncertainty matrix. It fails, however, for highly nonlinear systems. The failure motivated the development of the extended Kalman filter by which the nonlinear characteristics of the system's dynamics are approximated by a version of the system that is linearized around the last state estimate. The extended version has been popular due to its ability for handling nonlinear systems and non-Gaussian noise.

Evensen [14] identified a closure problem associated with the extended Kalman filter in the evolution equation for the error covariance. The problem in this context is having more unknowns than equations. The linearization used in the extended filter discards higher-order moments in the equation that governs the evolution of the error covariance. But, because this kind of closure technique produces an unbounded growth of error, the ensemble Kalman filter was introduced to alleviate the closure problem, which is a Monte Carlo method in which the model states and uncertainty are represented by an ensemble of realizations of the system [15].

The ensemble Kalman filter is conceptually simple and requires relatively low computation, which is why it has received increasing attention in history matching problems and continuous updating of models, as new data become available. Since instead of computing the state covariance using a recursive method, the method estimates the covariance matrix from a number of realizations, its computational cost is low. The ensemble Kalman filter has been shown to be very efficient and robust for real-time updating in various fields, such as weather forecasting [16], oceanography, and meteorology [17]. It was also used in the development of dynamic models of large-scale porous media [18] and optimizing gas production from large landfills [19], in both of which dynamic data become available over a period of time. The reader is referred to Ref. [19] for complete details of the method and how it is implemented.

B. Data assimilation and machine learning

When the governing equations for a complex phenomenon, which are in terms of ordinary or partial differential equations, are known and are solved numerically to describe the dynamic evolution of the phenomenon, uncertainties often remain and are usually of one of two types: (a) the internal variability that is driven by the sensitivity to the initial conditions, and (b) the errors generated by the model or the governing equations. The first type has to do with the amplification of the error in the initial condition, and arises even if the model is complete and "perfect." It is mitigated by using data assimilation, briefly described above. The second type has recently been addressed by use of machine-learning techniques, which have been emerging as an effective approach for addressing the issue of models' errors. To develop reduced-order models (ordinary or partial differential equations) for complex phenomena, the variables and scales are grouped into unresolved and resolved categories, and machine-learning approaches are emerging as being particularly suitable for addressing the errors caused by the unresolved scales.

To see the need for addressing the errors due to unresolved scales, consider, for example, the current climate models. The resolution of the computational grids used in the current climate models is around 50–100 km horizontally, whereas many of the atmosphere's most important processes occur on scales much smaller than such resolutions. Clouds, for example, can be only a few hundred meters wide, but they still play a crucial role in Earth's climate since they transport heat and moisture. Carrying out simulations with a resolution comparable with the size of such clouds is impractical for the foreseeable future. To address the issue, two approaches have been used to combine data assimilation with a machine-learning approach.

(i) The first approach is based on learning physical approximation, usually called subgrid parametrization, which are typically computationally expensive. Alternatively, the same can be achieved based on the differences between high- and low-resolution simulations. For climate models, for example, parametrizations have been heuristically developed over the past several decades and tuned to observations; see, for example, Hourdin *et al.* [20]. Due to the extreme complexity of the system, however, significant inaccuracies still persist in the parametrization, or physical approximations of, for example, clouds in the climate models, particularly given the

fact that clouds also interact with such important processes as boundary-layer turbulence and radiation. Given the debate over global warming and how much Earth will warm up as a result of increased greenhouse gas concentrations, the fact that such inaccuracies manifest themselves as model biases only goes to show the need for accurate and computationally affordable models.

(ii) In the second approach one attempts to emulate the entire model by using observations, and spatially dense and noise-free data. Various types of neural networks, including convolutional [21,22], recurrent [23], residual [24], and echo state networks [25] have been utilized. An echo state network is a *reservoir computer*, i.e., a computational framework based on the theory of recurrent neural network (RNN) that maps input data onto higher-dimensional computational space through the dynamics of a fixed and nonlinear system called a reservoir, which uses a RNN with a hidden layer of low connectivity. The connectivity and weights of hidden neurons are fixed and randomly assigned. Dedicated neural network architectures, combined with a data assimilation method are used [26] to address problem of partial and/or noisy observations.

As discussed by Rasp *et al.* [27], cloud-resolving models do alleviate many of the issues related to parameterized convection. Although such models also involve their own tuning and parametrization, the advantages that they offer over coarser models are very significant. But climate-resolving models are also computationally too expensive, if one were to simulate climate change over tens of years in real time. Rapid increase in the computational power is making it possible, however, to carry out "short"-time numerical simulations, with highly resolved computational grids, which cover up to a few years. It is here that machine-learning approaches have begun to play an important role in addressing the issue of inaccuracies and grid resolution, because neural networks can be trained by the results of the short-term simulations, and then be used for forecasting over longer periods of time.

1. Example 1: Representing subgrid processes in climate models using a machine-learning algorithm

A good example is the approach developed by Rasp *et al.* [27] for representing subgrid processes in climate models. They trained a deep neural network to represent all atmospheric subgrid processes in a climate model. The training was done based on learning from a multiscale climate model that explicitly took convection into account. Then, instead of using the traditional subgrid parametrizations, the trained neural network was utilized in the global general circulation model, which could interact with the resolved dynamics and other important aspects of the core model. Their approach is a physics-informed machine-learning (PIML) algorithm, which are those in which, in addition to providing a significant amount of data for training the network, some physical constraints are also imposed on the algorithms. The constraint in this case was the climate model. We will return in the PIML algorithms in the next section, and describe them in detail.

The base model that Rasp *et al.* utilized [27] was version 3.0 of the well-known superparameterized community atmosphere model (SPCAM) [28] in an aquaplanet setup. Assuming a realistic equator-to-pole temperature gradient, the sea temperature was held fixed, with a full diurnal cycle (a pattern that recurs every 24 h), but no seasonal variation. In superparameterization, a 2D cloud-resolving model is embedded in each grid column (which in Rasp *et al.* [27] was 84 km wide) of the global circulation model, which resolves explicitly deep convective clouds and includes parameterizations for small-scale turbulence and cloud microphysics. For the sake of comparison, Rasp *et al.* [27] also carried out numerical simulations using a traditional parametrization package, usually referred to as the controlled CAM (CTRLCAM) [27]. The model and package exhibit many typical problems associated with traditional subgrid cloud parametrizations, including a double intertropical convergence zone, and too much drizzle, but also missing precipitation extremes, whereas SPCAM contains the essential advantages of full three-dimensional cloud-resolving models that address such issues with respect to observations.

The neural network used [27] was a nine-layer deep, fully connected net with 256 nodes in each layer and 5×10^5 parameters that were optimized to minimize the mean-squared error between the network's predictions and the training targets. The advantages of the deep neural network are that they have lower training losses, and are more stable in the prognostic simulations. Simulations were carried out for a period of five years, after a one-year spin-up—the time taken for an ocean model to reach a state of statistical equilibrium under the applied forcing. In the prognostic global simulations, the neural network parametrization interacted freely with the resolved dynamics, as well as with the surface flux scheme.

In Fig. 1(a) the results for the mean subgrig heating, computed by SPCAM, CTRLCAM, and neural network-aided model, referred to as NNCAM, are shown. The results computed by the latter two models are in very good agreement, whereas those determined by simulating the CTRLCAM package produced a double peak, usually referred to as the intertropical convergence zone in climate models. The corresponding mean temperatures are shown in Fig. 1(b), with the same level of agreement between the results based on the SPCAM and NNCAM. The results for the radiative fluxes predicted by the NNCAM parametrization were also in close agreement with those of SPCAM for most of the globe, whereas the results produced by CTRLCAM had large differences in the tropics and subtropics caused by its aforementioned double-peak bias. Figure 2 presents the results for precipitation distribution, indicating once again the inability of CTRLCAM in producing the correct results, since the computed distribution exhibits too much drizzle and absence of extremes. However, the results computed by SPCAM and NNCAM are in good agreement, including the tails of the distribution.

In terms of speeding up the computations, NNCAM parametrization was about 20 times faster than SPCAM's. Moreover, the neural network does not become more expensive at prediction time, even if trained with higher-resolution training data, implying that the approach can scale with ease to neural networks trained with much more expensive 3D global cloud-resolved simulations.

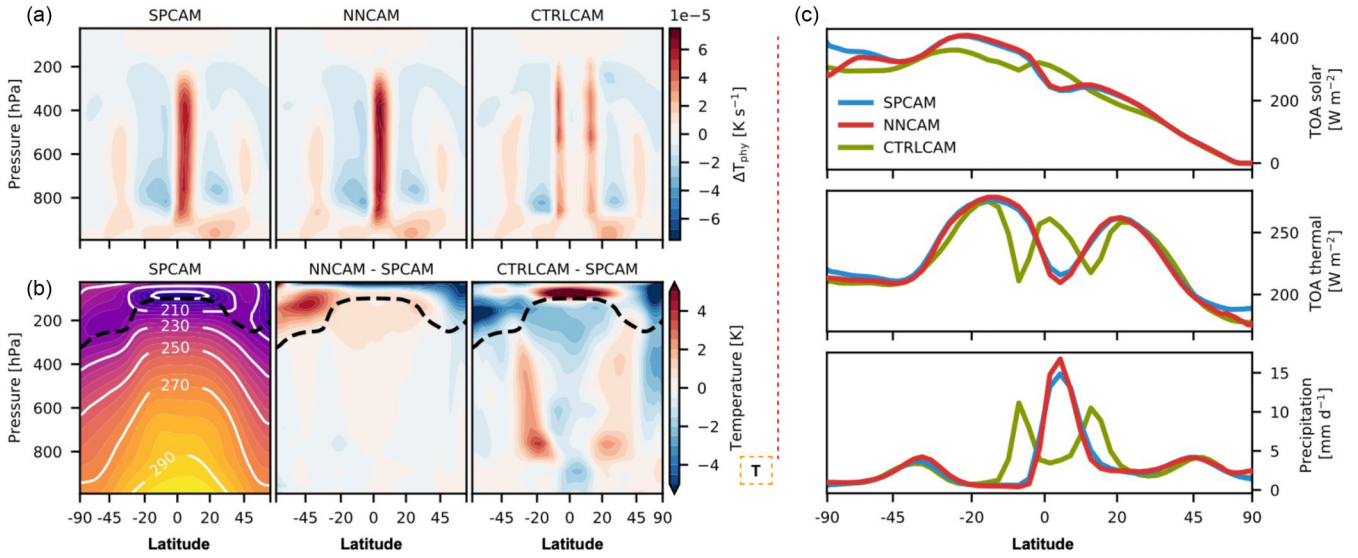

FIG. 1. Longitudinal and five-year temporal averages. (a) Mean convective and radiative subgrid heating rates ΔT_{phy}. (b) Mean temperature T of superparametrized community atmosphere model (SPCAM) and biases of neural network-aided community atmosphere model (NNCAM) and controlled community atmosphere model (CTRLCAM) relative to SPCAM. The dashed black line denotes the approximate position of the tropopause. (c) Mean short- (solar) and longwave (thermal) net fluxes at the top of the atmosphere and precipitation. The latitude axis is area weighted. Based on Ref. [27].

2. *Example 2: Inferring unresolved scale parametrization of an ocean-atmosphere model*

The second example that we briefly describe is the work of Brajard *et al.* [29], who developed a two-step approach in which one trains model parametrization by using a machine-learning algorithm and direct data. Their approach is particularly suitable for cases in which the data are noisy, or the observations are sparse. As the first step, a data assimilation technique was used, which was the ensemble Kalman filter (see above), to estimate the full state of the system based on a truncated model. The unresolved part of the truncated model was treated as model error in the data assimilation system. In the second step, a neural network was used to emulate the unresolved part, a predictor of model error given the state of the system, after which the neural network-based parametrization model was added to the physical core truncated model to produce a hybrid model.

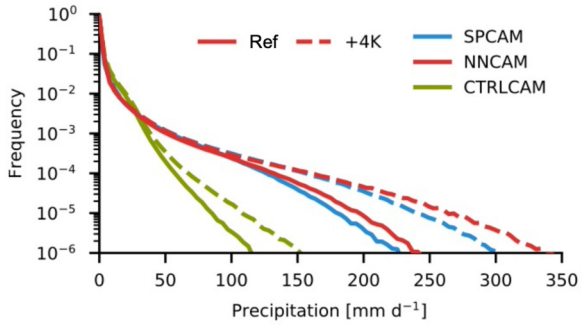

FIG. 2. Precipitation histogram of time-step (30 min) accumulation. The bin width is 3.9 mm d^{-1}. Solid lines show the simulations for reference sea surface temperature (SST). Dashed lines denote the simulation results for warming up by +4-K (see the original reference [30]). The neural network in the +4-K case is NNCAM-ref +4 K. Based on Ref. [27].

Brajard *et al.* [29] applied their approach to the modular arbitrary-order-ocean-atmosphere model (MAOOAM), which has three layers, two for the atmosphere and one for the ocean, and is a reduced-order quasigeostrophic model that is resolved in the spectral space. The model consists of N_a modes of the atmospheric barotropic streamfunction $\psi_{a,i}$ and the atmospheric temperature anomaly $T_{a,i}$, plus N_o modes of the oceanic streamfunction $\psi_{o,j}$ and the oceanic temperature anomaly $T_{o,j}$, so that the total number of variables is $N_x = 2(N_a + N_o)$. The ocean variables are considered as slow, while the atmospheric variables are viewed as the fast. Two versions of MAOOAM were considered, namely, the true model with dimension $N_a = 20$ and $N_o = 8$ ($N_x = 56$), and a truncated model with $N_a = 10$ and $N_o = 8$ ($N_x = 36$). The latter model does not contain 20 high-order atmospheric variables, ten each for the streamfunction and the temperature anomaly and, therefore, it does not resolve the atmosphere-ocean coupling that is related to high-order atmospheric modes.

The true model was used to simulate and generate synthetic data, part of which was used to train the neural network. It was simulated for a period of approximately 62 years after a spin-up of 30 000 years. The synthetic observations took into account the fact that observations of the ocean are not at the same scale as those of the atmosphere; thus, before being assimilated, instantaneous ocean observations were averaged over a 55 days rolling period centered at the analysis times. The architecture of the neural network was a simple three layers multilayer perceptrons.

To test the accuracy and predictive power, as well as the long-term properties of the two versions of MAOOAM and their hybrid with a neural network, three key variables, $\psi_{o,2}$, $T_{0,2}$ and $\psi_{a,1}$—the second components of ocean stream-

function and temperature and the first component of the atmospheric streamfunction—were computed, since they account, respectively, for 42, 51, and 18 percent of the variability of the models. Simulations of Brajard *et al.* [29] indicated that the predictions of the hybrid model, one consisting of data assimilation and the neural network with noisy data, matched very closely with the hybrid model with perfect data. In contrast, the truncated model's predictions differed from the true ones by a factor of up to 3.

We note that since many reconstruction methods involve computation of the derivatives of functions, which are difficult and may also introduce error, developing derivative-free methods is highly desirable. In this context, although ensemble Kalman-based systems have been successful, they cannot be systematically refined to return or produce the correct solution, except in the setting of linear forward models. Pavliotis *et al.* [31] proposed a reconstruction derivative-free approach to Bayesian reconstruction—an approach in which the data, the unknown parameter, and the noise in the observations are all treated as random variables, and one extracts information and assesses uncertainty based on the measured data, a model of the measurement process, and a model of *a priori* information—which may be used for posterior sampling or for maximum a posteriori estimation. The method relies on a fast and slow system of stochastic differential equations for the local approximation of the gradients.

Wider applications of the data assimilation combined with a machine-learning method do face challenges. For example, the computational architecture, such as multi-core supercomputers and graphics processing units, and the data types used for physics-based numerical simulation and for machine-learning algorithms, can be very different. Moreover, training and running hybrid models efficiently impose very heavy requirements on both the hardware and software. These are, of course, challanges for an emerging field.

C. Physics-informed machine-learning approaches

Although machine-learning algorithms and neural networks have been used for decades for predicting properties of various types of systems [32], the problem that many such algorithms suffer from is that they lack a rigorous, physics-based foundation and rely on correlations and regression. Thus, although they can fit very accurately a given set of data to some functional forms, they do not often have predictive power, particularly when they are tasked with making predictions for systems for which no data were "shown" to them, i.e., none or very little data for the properties to be predicted were used in training the neural network.

To address such a severe shortcoming, the PIML algorithms that were briefly mentioned above have been developed. As mentioned above, in PIML algorithms one, in addition to providing a significant amount of data for training the network, imposes some physics-based constraints on the algorithms. For example, if macroscopic properties of heterogeneous materials, such as their effective conductivity and elastic moduli, are to be predicted by a neural network, then, in addition to the data that are used for training it, one can also impose the constraint that the predictions must satisfy rigorous upper and lower bounds derived for the moduli [33,34]. Or, if one is to use a machine-learning algorithm to predict fluid flow and transport of a Newtonian fluid in a porous medium, then one can impose the constraint that the predictions must satisfy the Navier-Stokes equation, or the Stokes' equation if fluid flow is slow, and the convective-diffusion equation if one wishes to predict the concentration profile of a solute in the same flow field. Any other constraint that is directly linked with the physics of the phenomenon may also be imposed.

In general, three distinct approaches are being developed that contribute to the accuracy and acceleration of the training of a PIML algorithm that are as follows [4,5,7,35–37].

1. Multi-task learning

In this approach, the cost function to be minimized globally to develop the optimal machine-learning algorithm, and the neural network structure include the aforementioned constraints. In other words, it is not enough for the traditional cost function of the neural networks—the sum of the squares of the differences between the predictions and the data—to be globally minimum, but rather the cost function is penalized by imposing the constraints on it. Thus, the approach is a multi-task learning process, because not only the PIML algorithm is trained by the data, but the training also includes some physics-based constraints, such as a governing equation, upper and/or lower bounds to the properties of interest, and other rigorous information and insights, so that the predictions will also be based on, and satisfy, the constraints. The imposition of the constraints represents a bias in the training process, because they force the algorithm to be trained in a specific direction. We present two concrete examples to illustrate the method.

Example 1: Predicting fluid flow in a two-dimensional polymeric membrane. In this problem, a high-resolution three-dimensional (3D) image of a membrane of size $500 \times 500 \times 1000$ voxels was used [7], whose porosity, thickness, permeability, and mean pore size were known. Seven hundred 2D slices with a size 175×175 pixels were extracted from the 3D image, and fluid flow in the slices was simulated by solving the Navier-Stokes equations, with part of the results used in the training the algorithm.

A physics-informed recurrent encoder-decoder (PIRED) network was then developed. The network, a supervised one, consisted of encoder and decoder, known as the U-Net and residual U-Net (RU-Net), whose architecture is shown in Fig. 3. The encoder had four blocks, with each block containing the standard convolutional and activation layers, as well as pooling and batch normalization layers. The pooling layer compressed the input images to their most important features by eliminating the unnecessary ones, and stored them in the latent layer that consisted of the activation, convolutional, and batch normalization layers. The batch normalization layer not only allowed the use of higher learning rates by reducing internal covariate shift, but also acted as a regularizer for reducing overfitting [38]. The mean $\langle x \rangle$ and variance $\mathcal{V}^2(x)$ of batches of data x were computed in the bath normalization layer, and a new normalized variable y was defined by

$$y = \gamma \frac{x - \langle x \rangle}{\sqrt{\mathcal{V}^2(x) + \epsilon}} + \beta. \tag{4}$$

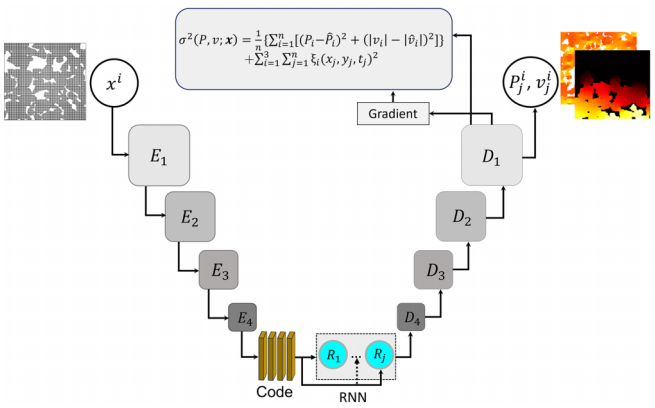

FIG. 3. The architecture of the PIRED network, with E_i and D_i indicating the encoder and decoder blocks; σ^2 being the cost function, and x^i the input. The pressure P^j and fluid velocity $|v|^j$ are the output. Based on Ref. [7].

Here, γ and β are learnable parameter vectors that have the same size as the input data, and ϵ is set at a typically small value, say 10^{-5}. During the training, the layer kept running estimates of its computed mean and variance, and utilized them for normalization during evaluation. The variance was calculated by the biased estimator.

The decoder also had four blocks. Each block contained the convolutional, activation and batch normalization layers, as well as a transposed convolutional layer that was similar to a deconvolutional layer in that, if, for example, the first encoder had a size $128 \times 64 \times 64$, i.e., 128 features with a size 64×64, then, the transposed convolutioonal layer in the decoder also had a similar size. The transposed convolutional layer utilized the features extracted by the pooling layer to reconstruct the output, which were the pressure and fluid velocity fields, P and $\mathbf{v}$, at various times. Because the latent layer of the recurrent neural network consisted of residual blocks, i.e., layers that, instead of having only one connection, were connected to more distant previous layers, it improved the performance of the PIRED, and sped up significantly the overall network's computations.

Assuming that the fluid is incompressible and Newtonian, the mass conservation equation for a 2D medium is given by $\boldsymbol{\nabla}\cdot\mathbf{v} = \partial v_x/\partial x + \partial v_y/\partial y = 0$, where both velocity components v_x and v_y and the spatial coordinates x and y are made dimensionless by a characteristic length L and characteristic velocity v_0. The (dimensionless) Navier-Stokes equation is given by

$$\frac{D\mathbf{v}}{Dt} = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\boldsymbol{\nabla}\mathbf{v} = -\boldsymbol{\nabla}P + \mathrm{Re}^{-1}\nabla^2\mathbf{v}, \tag{5}$$

where $\mathrm{Re} = \rho v_0 L/\mu$ is the Reynolds number, and $D/Dt = \partial/\partial t + \mathbf{v}\cdot\boldsymbol{\nabla}$ is the substantial derivative. Three residual functions, $\xi_1 = \boldsymbol{\nabla}\cdot\mathbf{v}$, $\xi_2 = Dv_x/Dt + \partial P/\partial x - \mathrm{Re}^{-1}\nabla^2 v_x$, and $\xi_3 = Dv_y/Dt + \partial P/\partial y - \mathrm{Re}^{-1}\nabla^2 v_y$, were defined and incorporated in the cost function σ^2, minimized by the PIRED network, instead of naively minimizing the squared differences between the data and predicted values of $\mathbf{v}$ and P.

The idea of incorporating the governing equations in the loss function was first proposed and implemented by Hamzehpour and Sahimi [39,40], who studied development of optimal models of large-scale porous media, given static and dynamic data for their various properties. To converge to the actual, numerically calculated values by solving the mass conservation and the Navier-Stokes equations, one must have, $\xi_i = 0$ for $i = 1$, 2, and 3. Thus, the PIRED network learned that the mapping between the input and output must comply with the requirement that, $\xi_i = 0$, which not only enriched its training, but also accelerated convergence to the actual values of P and $\mathbf{v}$. The cost function σ^2 was, therefore, defined by

$$\sigma^2 = \frac{1}{n}\left\{\sum_{i=1}^{n}[(P_i - \hat{P}_i)^2 + (|v_i| - |\hat{v}_i|)^2]\right\} + \sum_{i=1}^{3}\sum_{j=1}^{n}\xi_i(x_j, y_j, t_j)^2, \tag{6}$$

where n is the number of data points used in the training, and P_i and $|v_i|$ are the actual pressure and magnitude of the fluid velocity at point (x_i, y_i) at time t_i, with the superscript $\hat{}$ denoting the predictions by the PIRED network. The P and $\mathbf{v}$ fields were computed at four distinct times. Note that the amount of the data needed for computing P and $\mathbf{v}$ was significantly smaller than what would be needed by standard machine-learning methods.

A fluid was injected into the membrane at one side with a constant speed v_0, and extracted at the opposite side at a fixed pressure. The other two boundaries were assumed to be impermeable. Solving the mass conservation and the Navier-Stokes equations in each 2D image took about 6 CPU minutes. The computations for training the PIRED network on an Nvidia Tesla V100 graphics processing unit (GPU) took about 2 GPU hours. Then, the tests for accuracy took less than a second. Part of the results were used in the training, and the rest in testing and comparing with the predictions of the PIRED network.

The reverse Kullback-Leibler divergence (relative entropy) [41] was used to minimize the cost function σ^2. If $p(x)$ is the true probability distribution of the input and output data, and $q(x)$ is an approximation to $p(x)$, then the reverse Kullback-Leibler divergence from $q(x)$ to $p(x)$ is a measure of the difference between the two. The aim is, of course, to ensure that $q(x)$ represents $p(x)$ accurately enough that it minimizes the reverse Kullback-Leibler divergence $D_{\mathrm{KL}}(q\|p)$, defined by

$$D_{\mathrm{KL}}[q(x)\|p(x)] = \sum_{x\in S} q(x)\log\left[\frac{q(x)}{p(x)}\right], \tag{7}$$

where S is the space in which $p(x)$ and $q(x)$ are defined. $D_{\mathrm{KL}} = 0$, if $q(x)$ matches $p(x)$ perfectly and, in general, it may be rewritten as

$$D_{\mathrm{KL}}[q\|p] = \mathbb{E}_{x\sim q}[-\log p(x)] - H[q(x)], \tag{8}$$

where $H[q(x)] = \mathbb{E}_{x\sim q}[-\log q(x)]$ is the entropy of $q(x)$, with $\mathbb{E}$ denoting the expected value operator and, thus, $\mathbb{E}_{x\sim q}[-\log p(x)]$ being the cross-entropy between q and p. Optimization of D_{KL} with respect to q is defined by

$$\begin{aligned}\arg\min D_{\mathrm{KL}}[q\|p] &= \arg\min \mathbb{E}_{x\sim q}[-\log p(x)] - H[q(x)] \\ &= \arg\min \mathbb{E}_{x\sim q}[\log p(x)] + H[q(x)].\end{aligned} \tag{9}$$

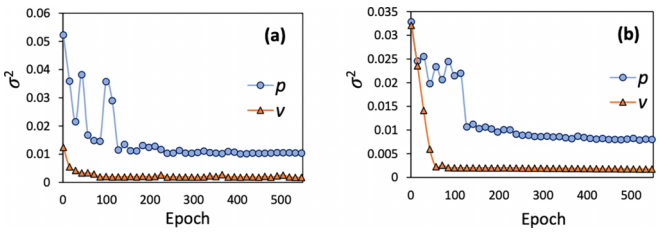

FIG. 4. Computational efficiency and accuracy of the PIRED. Comparison of the cost function σ^2 for the (a) training and (b) testing data for pressure p and fluid velocity v. Based on Ref. [7].

Thus, according to Eq. (9), one samples data points from $q(x)$ and does so such that they have the maximum probability of belonging to $p(x)$. The entropy term of Eq. (9) "encourages" $q(x)$ to be as broad as possible. The autoencoder tries to identify a distribution $q(x)$ that best approximates $p(x)$.

The trained PIRED network was used to reconstruct the velocity and pressure field in new (unused in training) 2D images using only a small number of images. Figures 4(a) and 4(b) present, respectively, the change in the cost function σ^2 for the training and testing datasets of the network. σ^2 decreases for both P and $\mathbf{v}$ during both the training and testing, indicating convergence toward the true solutions for both fields.

An effective permeability K was defined by, $K = \mu Lq/(A\Delta P)$, where q, A and ΔP are, respectively, the steady-state volume flow rate, and the surface area perpendicular to the macroscopic pressure drop ΔP. K was computed for 300 testing slices, and was predicted by the PIRED network as well. The comparison is shown in Fig. 5(a). But a most stringent test of the PIRED network is if it can predict accurately the permeability (and other properties) of a completely different porous medium without using any data associated with it. Thus, the image of a Fontainebleau sandstone [42] with a porosity of 0.14 was used. Since the sandstone's morphology is completely different from the polymeric membrane's, a slightly larger number of 2D slices from the membrane (not the sandstone) was utilized to better train the PIRED network. Figure 5(b) compares the effective permeabilities of one hundred 2D slices of the sandstone with the predictions of the PIRED network.

Example 2: Predicting arterial blood pressure in cardiovascular flows. Predictive modeling of cardiovascular flows and aspire is a valuable tool for monitoring, diagnosis, and surgical planning that can be utilized for large patient-specific topologies of systemic arterial networks, to obtain detailed predictions for, for example, wall shear stress and pulse wave propagation. The models that were developed in the past relied heavily on preprocessing and calibration procedures that require intensive computations, hence hampering their clinical applicability. Kissas *et al.* [43] developed a machine-learning approach, a physics-informed neural network (PINN) for seamless synthesis of noninvasive in-vivo measurements and computational fluid dynamics. In many ways, their PINN is similar to the PIRED described above, except that in Kamrava *et al.*'s work [7] the input data included a digitized image of the system.

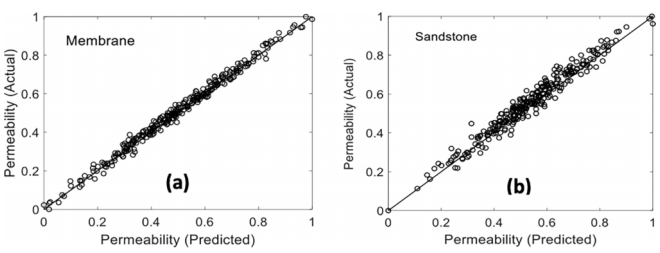

FIG. 5. Comparison of the actual permeabilities K and the predictions by the PIRED network for (a) 300 2D images of the membrane, and (b) for 100 images of the Fontainebleau sandstone. K is normalized according to $(K - K_{\min})/(K_{\max} - K_{\min})$. Based on Ref. [7].

Making a few assumptions, Kissas *et al.* [43] modeled pulse wave propagation in arterial networks by a reduced order (simplified) 1D model based on mass conservation and momentum equations,

$$\frac{\partial A}{\partial t} + \frac{\partial (A v_x)}{\partial x} = 0, \tag{10}$$

$$\frac{\partial v_x}{\partial t} + \alpha v_x \frac{\partial v_x}{\partial x} + \left(\frac{v_x}{A}\right)\frac{\partial}{\partial x}[(\alpha - 1)A v_x] + \frac{1}{\rho}\frac{\partial P}{\partial x} - K_R \frac{v_x}{A} = 0. \tag{11}$$

Here, $A(x,t)$, $v_x(x,t)$, and $P(x,t)$ denote, respectively, the cross-sectional area, blood's velocity, and pressure at time t, with x being the direction of blood flow; α is a momentum flux correction factor; ρ is the blood's density, and K_R is a friction parameter that depends on the velocity profile (flow regime). However, since the artery is an elastic deformable material, the constraint imposed by mass and momentum conservation is not sufficient for determining the pressure, since only the pressure gradient appears in the momentum equation. Assuming, however, that the artery is a linearly elastic material, the constitutive law for displacement of its walls, given by

$$P = P_e + \beta(\sqrt{A} - \sqrt{A_0}), \tag{12}$$

relates directly the arterial wall displacement to the absolute pressure in each cross section. Here, β is a coefficient related to the Young's modulus and the Poisson's ratio of the artery; $A_0 = A(x,0)$, and P_e is the external pressure. Thus, as another constraint, the constitutive relation was coupled to the mass and momentum conservation laws, implying that the correlations between them can be exploited through the PINN to determine the absolute pressure from velocity and cross-sectional area measurements. The system that Kissas *et al.* [43] modeled and studied, a Y-shaped bifurcation, is shown in Fig. 6. Three-dimensional geometries recovered from magnetic resonance imaging data and the corresponding centerlines (shown in Fig. 6) were extracted by using the vascular modeling toolkit library. The governing equations were then discretized and solved numerically by discontinuous Galerkin method, a numerical scheme that combines features of the finite-element and the finite-volume frameworks.

Thus, similar to the PIRED example described above, three residual functions, ξ_i with $i = 1, 2$, and 3, were defined by the left-hand sides of Eqs. (10)–(12). Several factors contribute to the overall cost, or loss, function, σ^2, which should be minimized globally. They are, (a) the usual sum of the

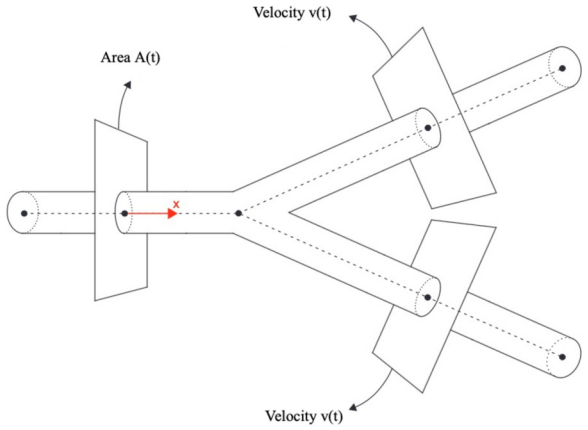

FIG. 6. Schematic representation of a Y-shaped bifurcating arterial system and its 1D centerlines used in the reduced model. Based on Ref. [43].

squared differences between the computed blood velocity and the arterial cross section and the corresponding data at every computational point (x, t). Blood velocity data are typically obtained using Doppler ultrasound or 4D flow MRI, while the area data are gleaned from 2D Cine images recovered by 4D flow MRI. (b) The sum of the squared residual functions ξ_i, defined above, for a sample of the collocation points used in the numerical simulation of mass conservation and momentum equation. (c) Contributions by the junctions at the bifurcation points shown in Fig. 6. We refer to the channel on the left-hand side of Fig. 6 as artery 1, and the two on the right that bifurcate from it as numbers 2 and 3. Conservation of mass requires that, $A_1 v_1 - (A_2 v_2 + A_3 v_3) = 0$, where, for convecience, we deleted the subscript x of the fluid velocities. Moreover, conservation of momentum implies that $p_1 + \frac{1}{2}\rho v_1^2 - (p_2 + \frac{1}{2}\rho v_2^2) = 0$ and $p_1 + \frac{1}{2}\rho V_1^2 - (p_3 + \frac{1}{2}\rho v_3^2) = 0$. Thus, three additional residual functions, ξ_i with $i = 4, 5$, and 6 were defined by the left-hand sides of the above equations, and the overall cost function σ^2 was the sum of the three types of contributions.

In many problems of the type we discuss here, there maybe an additional complexity: The order of magnitude of fluid velocity, cross-sectional area, and pressure are significantly different. For example, one has, $P \sim 10^6$ Pa, $A \sim 10^{-5}$ m^2, and $v_x \sim 10$ m/s. Such large differences give rise to a systematic numerical problem during the training of the PINN, since it affects severely the magnitude of the back-propagated gradients that adjust the neural network parameters during training. To address this issue, Kissas *et al.* [43] made the governing equations dimensionless by defining a characteristic length and a characteristic velocity, so that they all take on values that are $\mathcal{O}(1)$. They then normalized the input to have zero mean and unit variance, since as Glorot and Bengio [44] demonstrated, doing so mitigates the pathology of vanishing gradients in deep neural networks. The activation function that Kissas *et al.* [43] utilized was a hyperbolic tangent function, $\tanh x$.

Three neural networks, one for each artery in the Y-shape system, were used. Each of the networks had seven hidden layers with one hundred neurons per layer, followed by hyperbolic tangent activation function. Two thousand collocation points were used in the discontinuous Galerkin method for solving the discretized equations. Other details of the approach and the model are given in the original reference.

Figure 7 presents the results in Y-shaped bifurcation. Figure 7(a) compares the predicted velocity wave, computed by discontinuous Galerkin solution, with the predictions of the PINN with and without nondimensionalization, while Fig. 7(b) does the same for the pressure. They were computed at the middle point of artery 1. The agreement is excellent. The same type of approach was utilized by Zhu *et al.* [45] for surrogate modeling and quantifying uncertainty, and by Geneva and Zabaras [46] and Wu *et al.* [47] for modeling of nonlinear dynamical systems.

The works of Karmrava *et al.* [7] and Kissas *et al.* [43], as well as those of others [45–47], are representative of the PIML approach. The general approach is not restricted to problems described above involving numerical simulation of flow and transport equations. As mentioned above, Hamzehpour and Sahimi [39,40] were the first to incorporate the governing equations in the loss function to develop an optimal model of a large-scale porous medium. Part of the dynamic data that they used to construct the model and minimizing the loss function were seismic records, for which they incorporated the governing equation for wave propagation. In addition, suppose, for example, that in the PIRED example, fluid flow is not isothermal. Thus, one must incorporate the energy equation governing the temperature T of the system, $DT/Dt = \partial T/\partial t + (\mathbf{v}\cdot\nabla)T = \alpha_T \nabla^2 T$, where α_T is the thermal diffusivity. Clearly, then, a fourth residual function,

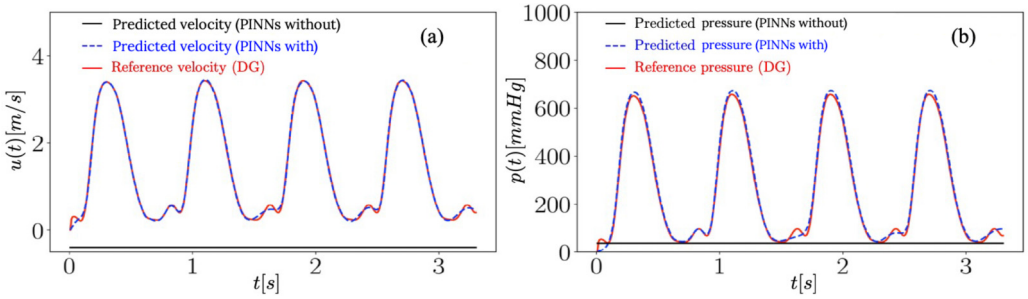

FIG. 7. Flow through a Y-shaped bifurcation. (a) Comparison of the computed velocity, $u(t) = v_x(t)$ wave, obtained by the discontinuous Galerkin (DG) method with those predicted by the PINN with and without nondimensionalization at the middle point of channel 1 (the left channel in Fig. 6). (b) Same as in panel (a) but for the pressure wave. Based on Ref. [43].

$\xi_4 = DT/Dt - \alpha_T \nabla^2 T$, should be incorporated in the loss functions. Similarly, if one is interested in, for example, mass transfer in a binary fluid mixture of A and B in the membrane (which is when a membrane is used for separation of fluid mixtures), then, given that C_A, the concentration of A, satisfies the mass continuity equation for a binary mixture, namely, $DC_A/Dt = \partial C_A/\partial t + \mathbf{v} \cdot \nabla C_A = D_{AB}\nabla^2 C_A$ (reaction terms may be added to the right-hand side, if there are chemical reactions), where D_{AB} is the mass diffusivity of A in a mixture of A and B, and $\mathbf{v} \cdot \nabla C_A$ represents a drift (convective) term, then, another residual function, $\xi_5 = DC_A/Dt - D_{AB}\nabla^2$ may be defined and incorporated in the loss function. If the mixture contains more than two components, then one defines other residual functions based on the Maxwell-Stefan equations for a multicomponent mixture.

Therefore, more generally, the loss function (6) should be written as

$$\sigma^2 = \frac{1}{n}\left\{\sum_{i=1}^{n}[(\mathcal{P}_i - \hat{\mathcal{P}}_i)^2]\right\} + \sum_{i=1}^{m}\sum_{j=1}^{n}\xi_i(\mathbf{x}_j, t_j)^2,$$

where $\mathcal{P}_i$ is the data point for property $\mathcal{P}$ at point i, $\hat{\mathcal{P}}_i$ is the corresponding computed property computed by the PINN, and m is the number of additional constraints ξ_j, such as the governing equations, imposed on the neural network algorithm. The advantage that the PIRED approach offers is that, if the input data include images, its encoder compresses the image to its essential features, hence saving a considerable amount of computation time.

Since such approaches involve numerical simulation of the governing equations imposed on the cost function of the neural networks, and in particular convolutional ones, an important question to address is the correct implementation of the boundary conditions in the learning process, which is particularly important in the PINN approach. As is well-known, there are three types of boundary conditions (BCs), namely, the Dirichlet BC—specifying the value of the unknown at the boundary—Neumann BC—specifying the flux)—and Robin BC (a mix of the first two). To address the problem, Sukumar and Srivastava [48] introduced *geometry-aware* trial functions to improve the training for partial differential equations by using concepts from constructive solid geometry, from which they utilized the so-called real functions, dubbed R-functions, which are used for representing and controlling the shape of a system, and generalized barycentric coordinates—a coordinate system in which the location of a point is specified by reference to a simplex, which is a triangle for points in 2D and a tetrahedron for points in 3D space—to construct an approximate distance function $\zeta(\mathbf{x})$ to the boundary of the domain in $\mathbb{R}^d$. To impose the homogeneous Dirichlet BC, the trial function was $\zeta(\mathbf{x})$ multiplied by the PINN approximation. Sukumar and Srivastava [48] used transfinite interpolation (a method for constructing functions over a planar domain such that they match a given function on the boundary) to generalize the trial function to *a priori* satisfy inhomogeneous Dirichlet, Neumann, and Robin BCs. This eliminated modeling error due to satisfying the BCs in a collocation method and, therefore, simplifies the training of the neural network.

2. *Learning aided by physical observations*

The training of any machine-learning algorithm can be improved by feeding it, as the input, observational data that convey the physics of the system under study. As mentioned in the Introduction, vast amounts of data are being collected for various complex phenomena. Thus, if such data, which provide insights into the phenomena, are used as the input to training of a machine-learning algorithm, they will bias it toward satisfying the observational data, implying that the final machine-learning tool should be capable enough for providing accurate predictions for those aspects of the phenomenon for which no data were fed to the algorithm as the input; see, for example, Kashefi *et al.* [49] who developed a point-cloud deep-learning algorithm for predicting fluid flow in disordered media. A point cloud is a set of data points that is typically sparse, irregular, orderless, and continuous, encodes information in 3D structures, and is in per-point features that are invariant to scale, rigid transformation, and permutation. Due to such characteristics, feature extractions from a point cloud is difficult for many deep-learning models.

3. *Embedding prior knowledge and inductive biases*

One may design neural networks in which prior knowledge and inductive biases are embedded, to facilitate making predictions for the phenomena of interest. There are several procedures that can be used for this purpose.

Convolutional neural networks. They were first proposed by LeCun *et al.* [50], and are considered to be the best known examples of such approaches. They were originally designed such that the invariance along groups of symmetries and patterns found in nature were honored. It has also been possible to design more general convolutional neural networks that honor such symmetry groups as rotations and reflections, hence leading to the development of architectures that depend only on the intrinsic geometry, which have been shown to be powerful tools for analyzing medical images [51] and climate pattern segmentation [52].

Kernel methods. Such methods [53], which minimize the cost function over a space of functions, rather than over a set of parameters as in the old neural network, represent a class distinct approaches that fall into the category of algorithms that improve the performance of the PIML approaches. They were motivated [53–55] by the physics of the systems under study. Moreover, many approaches that utilize neural networks have close asymptotic links to the kernel methods. For example, Wang *et al.* [56,57] showed that the training dynamics of the PIML algorithms can be understood as a kernel regression method in which the width of the network increases without bound. In fact, neural network-based methods may be rigorously interpreted as kernel methods in which the underlying *warping kernel*—a special type of kernels that were initially introduced [58] to model nonstationary spatial structures—is also learned from data.

Graph neural networks. In many machine-learning processes, the training process must deal with data that are presented as graphs that contain relations and correlations between the information. Examples include learning molecular fingerprints, protein interface, classifying diseases, and reasoning on extracted structures, such as the dependency

trees of sentences. Graph neural networks and their variants, such as graph convolutional networks, graph attention networks, and graph recurrent networks, have been proposed for such problems, and have proven to be powerful tools for many deep-learning tasks. An excellent review was given by Zhou *et al.* [59]; see also Refs. [7,36,37] for their applications.

It should be clear that one may combine any of the above three approaches to gain better performance of machine-learning algorithms. In addition, as the PIRED example described above demonstrated, when one deals with problems involving fluid flow, transport, and reaction processes in heterogeneous media, one may introduce dimensiolness groups, such as the Reynolds, Froude, and Prandtl numbers that not only contain information about and insights into the physics of the phenomena, but may also help one to upscale the results obtained by the PIML algorithm to larger length and timescales.

The field of PIML algorithms has been rapidly advancing. Many applications have been developed, particularly for problems for which either simulations based on advanced classical numerical simulations pose extreme difficulty, or they the problems so ill-posed that render the classical methods useless. They include, in addition to those referenced above, PIML for 4D flow magnetic resonance imaging data [43], predicting turbulent transport on the edge of magnetic confinement fusion devices—a problem that has been studied for several decades [60]—and a fermionic neural network (dubbed FermiNet) for ab initio computation of the solution of many-electron Schrödinger equation [61–63], which is a hybrid approach for informing the neural network about the physics of the problem. Since the wavefunctions must be parameterized, a special architecture was designed for FermiNN that followed the Fermi-Dirac statistics, i.e., it was anti-symmetric under the exchange of input electron states and the boundary conditions. As such, the parametrization was a physics-informed process. FermiNet was also trained by a physics-informed approach in that, the cost function was set as a variational form of the value of the energy expectation, with the gradient estimated by a Monte Carlo method. Several papers have explored application of the PIML to geoscience [4,7,35–37,63–67], as well as to large-scale molecular dynamics simulations [68] in which a neural network is used to represent the potential energy surfaces, and preprocessing is used to preserve the translational, rotational and permutational symmetry of the molecular system. The algorithm can be improved by using deep potential molecular dynamics, DeePMD [69], which makes it possible to carry out molecular dynamics simulations with one hundred million atoms for more than one nanosecond long [67], as well as simulations whose accuracy was comparable with ab initio calculations with one million atoms [70,71].

IV. TYPE-II SYSTEMS

Recall that in this type of systems data for a complex phenomenon in a heterogeneous system are available, and the physics of the phenomena of interest is also at least partially known. This type of systems may be divided into two groups. (a) The governing equations contain coefficients that must be evaluated based on the available data, because such coefficients depend on the morphology of the system. For example, any fluid flow and transport phenomenon in heterogeneous porous media is governed by the equations that describe mass and momentum conservation, but they contain flow and transport coefficients, such as the permeability, diffusivity, and the dispersion coefficients, which must be evaluated based on the available data. (b) The available data are for a complex phenomenon that exhibits multiscale features, in which case reconstructing the complete governing equations based on the data is a very difficult task that involves prohibitive computations. In such a case one resorts to a PIDD dimensionality reduction technique, i.e., discovering an ordinary or partial differential equation with the lowest order that describes the data.

A. Machine-learning approach

One way of determining the coefficients that appear in the governing equations for a complex phenomenon in a heterogeneous system is using a machine-learning algorithm to link the structure of the system to the coefficients. THe same approach can also be used to discover some terms of the governing equations that are required for solving the governing equations.

1. Example: Linking the permeability of a porous medium to its morphology

Slow flow through a porous medium is governed by the Stokes' equation, together with Darcy's law, $\mathbf{v} = -(K_e/\mu)\nabla P$, that relates the fluid's velocity $\mathbf{v}$ to the pressure gradient ∇P, where K_e is the effective permeability of the pore space, and μ is the fluid's viscosity. A long-standing problem has been the relationship between K_e and the morphology of porous media [9]. Various approaches have been proposed to address the problem [9,72], based on various approximations.

Kamrava *et al.* [35] developed a network that utilized a DL algorithm to link the morphology of porous media to their effective permeability. The network was neither a purely traditional artificial neural network, nor was it a purely DL algorithm, but rather a hybrid of both. Ten 3D x-ray images of actual sandstones were utilized as the input data. By computing the porosity of the samples of various sizes, images of size 200^3 voxels were selected as the representative elementary volume of the core. The input data were preprocessed to prepare them for the convolutional neural network (CNN), which was transforming the initial log-normal distribution of the permeabilities to a Gaussian distribution, since a CNN can better connect the identified features, when the distribution is Gaussian. To carry out the transformation, the PDF of the permeability data was constructed based on which the cumulative density function (CDF) was computed. The target PDF, namely, the Gaussian distribution was also constructed using the mean and variance of the data, after which the corresponding CDF was determined. Having the two CDFs, the new PDF and, consequently, the new permeability values were calculated. To do so, for each selected permeability from the original CDF graph its equivalent permeability from the target CDF was determined, i.e., the Gaussian distribution.

Since the ten samples were not nearly enough to provide the required variability in the types of morphology that one encounters in sandstones, the number of 3D images was increased using two methods. One was the Boolean method by which, given the statistical distribution of the sizes of solid objects, one produces many realizations of their packing. One hundred of such realizations of each type of the packing were generated that covered a wide range of pore-size distribution. In the second approach the original ten 3D x-ray images were used to generate 500 stochastic realizations of them, 50 for each of them, using the cross-correlation based simulation [73–77]. All the realizations complied with the porosity distribution of the original 3D image. Thus, the DL algorithm used an enriched database that contained realizations of pore space with diverse morphologies.

The algorithm that was used was a supervised learning, i.e., all the training inputs had their own label, which was their permeability. The network had the usual convolutional, activation, pooling, and fully connected layers. The overall procedure was as follows:

(i) The number of filters and their sizes, the architecture of the network, and other parameters were set.

(ii) The weights, biases, and filter matrices were initialized, with the first two initialized by selecting their values from a Gaussian distribitioon with zero mean and a unit variance.

(iii) The images of porous media were supplied to the network. The input was processed by various layers of the CNN, after which it produced its first estimate of the output, the permeabilities.

(iv) The cost function was computed. If it was larger than a preset threshold, then back-propagation was used to calculate the gradients of the error with respect to all the variables, and stochastic gradient descent was utilized to update all the weights and filter values and other parameter to minimize the error.

(v) Steps (iii) and (iv) were repeated for all the training data until the error reached a plateau and did not change any more.

Kamrava *et al.* [35] demonstrated that the network successfully developed accurate correlations between the morphology of porous media and their effective permeability. The high accuracy of the network was demonstrated by its predictions for the permeability of a variety of porous media.

A similar deep-learning algorithm was used to link the morphology of porous media to the dispersion coefficient of a solute transported by slow flow of a solvent through the same pore space [64]. Others used similar ideas to link the effective diffusivity [65] and other propertties [78,79] to the morphology of porous media. The same type of approaches have been used for developing a mapping between the hydraulic conductivity field of a porous formation and the macrodispersion coefficient in a large-scaler porous medium represented by a 2D Gaussian field [66].

B. Data-driven approach for reconstructing Kramers-Moyal expansion

The approach has been developed for systems for which the data are in the form of nonstationary time series $X(t)$, or spatially varying series $X(\mathbf{x})$. Characterizing such nonstationary time and spatial series has been a problem of fundamental interest for a long time, as they are encountered in a wide variety of problems, ranging from economic activity [80], to seismic time series [81], heartbeat dynamics [82,83], and large-scale porous media [9], and their analysis has a long and rich tradition in the field of nonlinear dynamics [84–86]. Much of the effort has been focused on addressing the question of how to extract a deterministic dynamical system of equations by an accurate analysis of experimental data since, if successful, the resulting equations will yield all the important information about and insights into the system's dynamical properties.

The standard approach has been based on treating the fluctuations in the data as stochastic variables that have been superimposed *additively* on a trajectory or time series that the deterministic dynamical system generates. The approach was originally motivated by the efforts for gaining deeper understanding of turbulent flows [87,88], and has been evolving ever since. Although it has already found many applications [89], it is still under further development (see below). More importantly, the approach has demonstrated the necessity of treating the fluctuations in the data as dynamical variables that interfere with the deterministic framework.

In this approach, given a nonstationary series $X(t)$, one constructs a stationary process $y(t)$, which can be done by at least one of two methods. (a) The *algebraic increments*, $y(t) = X(t+1) - X(t)$, are constructed. The best-known example of such series is the fractional Brownian motion (FBM) [90] with a power spectrum, $S(\omega) \propto 1/\omega^{2H+1}$, where H is the Hurst exponent. It is well-known that the FBM's increments, with $S(\omega) \propto 1/\omega^{2H-1}$, called fractional Gaussian noise [90], are stationary. Moreover, when $H = 1/2$, the increments are uncorrelated, whereas for $H = -1/2$ $X(t)$ becomes random. (b) Let $Z = \ln X(t)$. Then, one constructs the *returns* $y(t)$ of $X(t)$ defined by, $y(t) = Z(t+1) - Z(t) = \ln[X(t+1)/X(t)]$, so that $y(t)$ is the *logarithmic increments* series. It is straightforward to show that both approaches produce stationary series by studying their various moments over windows of different sizes in the series. One then analyzes $y(t)$ based on the application of Markov processes and derives a governing equation for the series based on a Langevin equation, the details of which are as follows.

One first checks whether $y(t)$ does follow a Markov chain [91,92]. If so, then its Markov timescale t_M—the minimum time interval over which $y(t)$ can be approximated by a Markov process—is estimated (see below). In general, to characterize the statistical properties of any series $y(t)$, one must evaluate the joint probability distribution function $P_n(y_1, t_1; \cdots; y_n, t_n)$ for the number of the data points, n. If, however, $y(t)$ is a Markov process, then the n-point joint probability distribution function P_n is given by

$$P_n(y_1, t_1; \cdots; y_n, t_n) = \prod_{i=1}^{n-1} P(y_{i+1}, t_{i+1}|y_i, t_i),$$

where $P(y_{i+1}, t_{i+1}|y_i, t_i)$ is the conditional probability. Moreover, satisfying the Chapman-Kolmogorov equation [93],

$$P(y_2, t_2|y_1, t_1) = \int dy_3\, P(y_2, t_2|y_3, t_3) P(y_3, t_3|y_1, t_1), \quad (13)$$

is a necessary condition for $y(t)$ to be a Markov process for any $t_3 \in (t_1, t_2)$. (The opposite is not necessarily true, namely, if a stochastic process satisfies the Chapman-Kolmogorov equation, it is not necessarily Markov). Therefore, one checks the validity of the Chapman-Kolmogorov equation for various values of y_1 by comparing the directly evaluated $P(y_2, t_2|y_1, t_1)$ with those computed according to right-hand side of Eq. (13).

The Markov timescale t_M may be evaluated by the least-squares method. Since for a Markov process one has

$$P(y_3, t_3|y_2, t_2; y_1, t_1) = P(y_3, t_3|y_2, t_2), \tag{14}$$

one compares $P(y_3, t_3; y_2, t_2; y_1, t_1) = P(y_3, t_3|y_2, t_2; y_1, t_1)P(y_2, t_2; y_1, t_1)$ with that obtained based on the assumption of $y(t)$ being a Markov process. Using the properties of Markov processes and substituting in Eq. (14) yield

$$P_{\rm M}(y_3, t_3; y_2, t_2; y_1, t_1) = P(y_3, t_3|y_2, t_2)P(y_2, t_2; y_1, t_1). \tag{15}$$

One then computes the three-point joint probability distribution function through Eq. (14) and compares the result with that obtained through Eq. (15). Doing so entails, first, determining the quality of the fit by computing the least-squares fitting quantity χ^2, defined by

$$\chi^2 = \int dy_3\, dy_2\, dy_1 \Bigg[P(y_3, t_3; y_2, t_2; y_1, t_1) - \frac{1}{\sigma_{3j}^2 + \sigma_{\rm M}^2} P_{\rm M}(y_3, t_3; y_2, t_2; y_1, t_1) \Bigg]^2, \tag{16}$$

where σ_{3j}^2 and $\sigma_{\rm M}^2$ are, respectively, the variances of $P(y_3, t_3; y_2, t_2; y_1, t_1)$ and $P_{\rm M}(y_3, t_3; y_2, t_2; y_1, t_1)$. Then, t_M is estimated by the likelihood statistical analysis. In the absence of a prior constraint, the probability of the set of three-point joint probability distribution functions is given by

$$P(t_3 - t_1) = \Pi_{y_3, y_2, y_1} \frac{1}{\sqrt{2\pi\left(\sigma_{3j}^2 + \sigma_{\rm M}^2\right)}} \exp\left\{ \frac{[P(y_3, t_3; y_2, t_2; y_1, t_1) - P_{\rm M}(y_3, t_3; y_2, t_2; y_1, t_1)]^2}{2\left(\sigma_{3j}^2 + \sigma_{\rm M}^2\right)} \right\}, \tag{17}$$

which must be normalized. Evidently, then, when for a set of the parameters $\chi_\nu^2 = \chi^2/N$ is minimum (with N being the degree of freedom), the probability is maximum. Thus, if χ_ν^2 is plotted versus $t_3 - t_2$, then t_M will be the value of $t_3 - t_1$ at which χ_ν^2 is minimum [94].

Knowledge of $P(y_2, t_2|y_1, t_1)$ for a Markov process $y(t)$ is sufficient for generating the entire statistics of $y(t)$, which is encoded in the n-point probability distribution function that satisfies a master equation, which itself is reformulated by a Kramers-Moyal expansion [95],

$$\frac{\partial P(y, t|y_0, t_0)}{\partial t} = \sum_k (-1)^k \frac{\partial^k}{\partial y^k}[D^{(k)}(y, t)P(y, t|y_0, t_0)]. \tag{18}$$

The Kramers-Moyal coefficients $D^{(k)}(y, t)$ are computed by

$$D^{(k)}(y, t) = \frac{1}{k!} \lim_{\Delta t \to 0} M^{(k)},$$
$$M^{(k)} = \frac{1}{\Delta t} \int dy'(y' - y)^k P(y', t + \Delta t|y, t). \tag{19}$$

For a general stochastic process, all the coefficients can be nonzero. If, however, $D^{(4)}$ vanishes or is small compared to the first two coefficients [93], then truncation of the Kramers-Moyal expansion after the second term is meaningful in the statistical sense, in which case the expansion is reduced to a Fokker-Planck equation that, in turn, according to the Ito calculus [93,94] is equivalent to a Langevin equation, given by

$$\frac{dy(t)}{dt} = D^{(1)}(y) + \sqrt{D^{(2)}(y)}\, \eta(t), \tag{20}$$

where $\eta(t)$ is a random "force" with zero mean and Gaussian statistics, δ-correlated in t, i.e., $\langle \eta(t)\eta(t')\rangle = 2\delta(t - t')$.

The Langevin equation makes it possible to reconstruct a time series for $y(t)$ similar, *in statistical sense*, to the original one, and can be used to make predictions for the future, i.e., given the state of the system at time t, one determines the probability of finding the system in a particular state at time $t + \tau$ by writing $X(t + 1)$ in terms of $X(t)$ by

$$X(t + 1) = X(t) \exp\{\sigma_y[y(t) + \bar{y}]\}, \tag{21}$$

where $\bar{y}$ and σ_y are the mean and standard deviations of $y(t)$. To use Eq. (21) to predict $X(t + 1)$, one needs $[X(t), y(t)]$. Thus, three consecutive points in the series $y(t)$ are selected and a search is carried out for three consecutive points in the reconstructed $y(t)$ with the smallest difference with the selected points. Wherever this happens is taken to be the time t which fixes $[X(t), y(t)]$.

1. Example 1: Fluctuations in human heartbeats

It has been shown that various stages of sleep may be characterized by extended correlations of heart rates, separated by a large number of beats. The method described above based on the Markov timescale t_M and the drift and diffusion coefficients, $D^{(1)}$ and $D^{(2)}$, provides crucial insights into the difference between the interbeat fluctuations of healthy subjects and patients with congestive heart failure. Figures 8 and 9 present [92,96] the drift and diffusion coefficients for the two groups of patients (for details of the data see the original references). In particular, the diffusion coeffcients of the healthy subjects and those with congestive heart failure are completely different. The important point to emphasize is that, the approach can detect such differences even at the earliest stages of development of congestive heart failure [92,93], when no other analysis can.

Despite its success, the approach is still under development. According to the Pawula theorem [97], only three

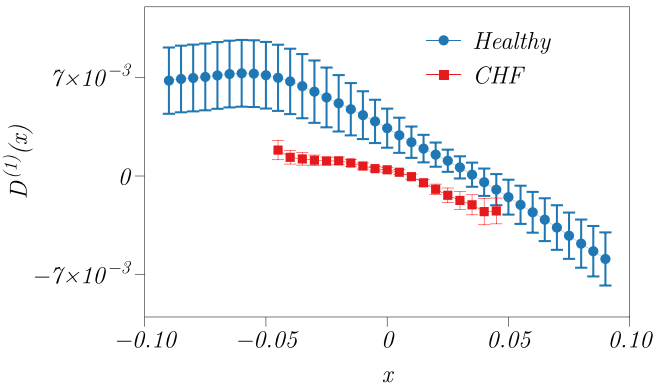

FIG. 8. The drift coefficient $D^{(1)}(x)$ for two classes of patients, the healthy ones and those with congestive heart failure (CHF).

outcomes are possible in a Kramers-Moyal equation of order k: (a) The expansion is truncated at $k = 1$, implying that the process is deterministic. (b) The expansion is truncated at $k = 2$, which results in the Fokker-Planck equation describing a diffusion process, and (c) the expansion must, in principle, contain all the terms, $k \to \infty$, in which case any truncation at a finite order $k > 2$ would produce a nonpositive probability distribution function. More importantly, it has become evident [98] that a nonvanishing $D^{(4)}(X,t)$, i.e., when the Kramers-Moyal expansion cannot be truncated after the second term, represents a signature of a jump discontinuity in the time series, in which case one needs the Kramers-Moyal coefficients of at least up to order six, i.e., up to $D^{(6)}(X,t)$, and in many cases even up to order eight [98], to estimate the jump amplitude and rate. For nonvanishing $D^{(4)}(X,t)$, the governing equation for a time series $X(t)$ with the jump-diffusion process is given by [30,98]

$$dX(t) = D^{(1)}(X,t)dt + \sqrt{D^{(2)}(X,t)}\,\eta(t) + \xi dJ(t), \quad (22)$$

where $J(t)$ is a Poisson jump process. The jump's rate $\lambda(x,t)$ can be state-dependent with a size ξ, and is given by, $\lambda(x,t) = M^{(4)}(x,t)/[3\sigma_\xi^4(x,t)]$, where, $\sigma_\xi^2(x,t) = M^{(6)}(x,t)/[5M^{(4)}(x,t)]$. Dynamic processes with jumps are highly important, as they have been used to describe random evolution, for example, of neuron dynamics [28,99],

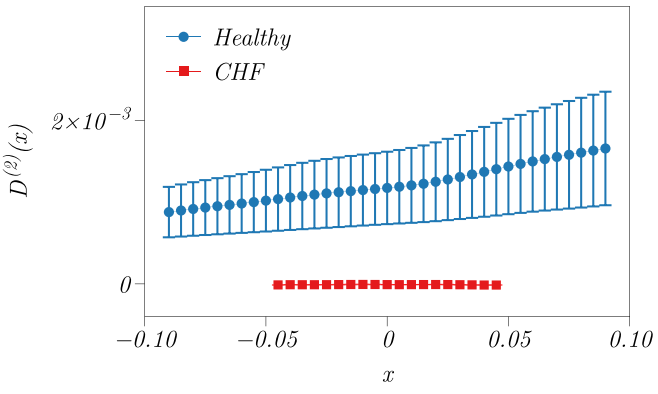

FIG. 9. The diffusion coefficient $D^{(2)}(x)$ for two classes of patients, the healthy ones and those with congestive heart failure (CHF).

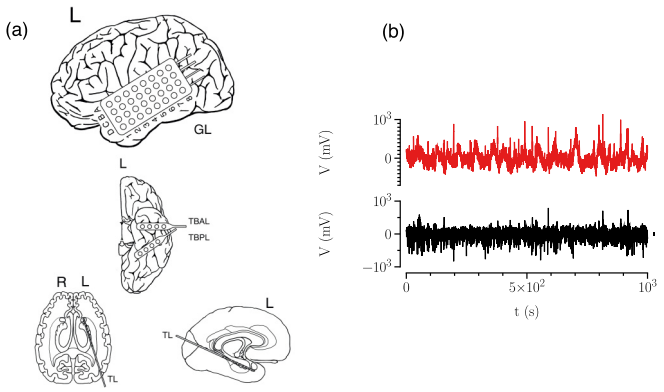

FIG. 10. (a) Implantation scheme of intracranial electrodes in a patient with seizures originating in the left mesial temporal lobe: temporal-lateral grid electrode (8 × 4 contacts, denoted by GL), two temporal-basal strip electrodes (4 contacts each, denoted by TB), and a hippocampal depth electrode (10 contacts, denoted by TL). The most anterior contact (TL1) is located ventral to the amygdala, while the most posterior contact (TL10) is located within the hippocampus. The latter electrode samples the epileptic focus. (b) Segments of the intracranial electroencephalographic (iEEG) time series recorded during the seizure-free interval from within the epileptic focus (contact TL4) and from a distant brain region (contact GLC6). Based on Ref. [98].

soil moisture dynamics [100], and such financial features as stock prices, market indices, and interest rates [101], and epileptic brain dynamics [98]. Let us describe a practical application of dynamic processes with jumps that is data-based and reconstruct the governing equation for the dynamics.

2. Example 2: Reconstruction of stochastic dynamics of epileptic brain

Brain's electrical rhythms in epileptic patients tend to become imbalanced, giving rise to recurrent seizures. When a seizure happens, the normal electrical pattern is disrupted by sudden and synchronized bursts of electrical energy that may briefly affect the consciousness of the patient, as well as the movements or sensations. Figure 10(b) presents intracranial electroencephalographic (iEEG) time series in a patient with seizures originating in the left mesial temporal lobe.

The first- and second-order Kramers-Moyal coefficients and Langevin-type modeling of iEEG time series can be used to construct stochastic qualifiers of epileptic brain dynamics, which yield valuable information for diagnostic purposes. In particular, it has been shown [102] that qualifiers based on the diffusion coefficient make it possible to obtain a more detailed characterization of spatial and temporal aspects of the epileptic process in the affected, as well as nonaffected brain hemispheres. There is, however, a major difference between the dynamics of the affected and nonaffected regions of brain, with the former region, responsible for the generation of focal epileptic seizures, being characterised by a nonvanishing fourth-order Kramers-Moyal coefficient, whereas that is not the case for the dynamics of latter region [102]. Thus, pathological brain dynamics is not described by continuous

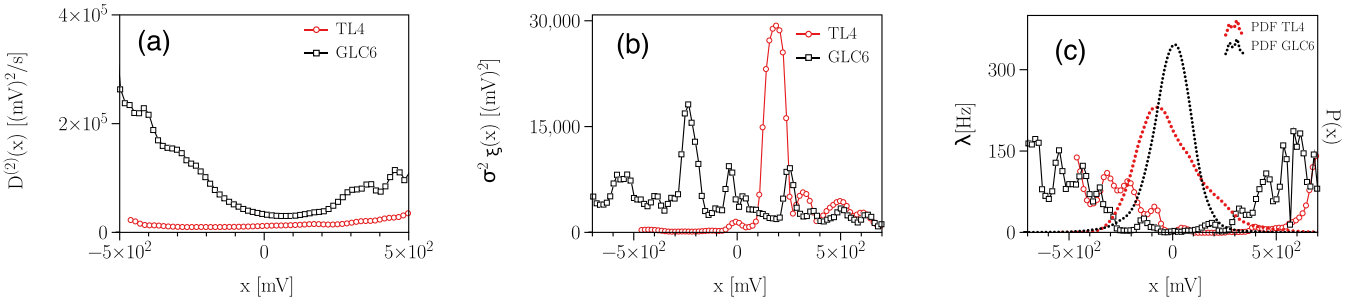

FIG. 11. Sample results computed based on the data for an epilepsy patient with an epileptic focus in the left mesial temporal lobe, showing the (a) diffusion coefficients, (b) jump amplitudes, and (c) jump rates, together with the respective probability distribution functions estimated from normalized iEEG time series, recorded during the seizure-free interval from within the epileptic focus (contact TL4 in Fig. 10) and the data from a distant brain region (contact GLC6, as shown in Fig. 10). Based on Ref. [98].

diffusion processes and, hence, by the Langevin-type modeling described above. Pathological iEEG time series exhibit highly nonlinear properties [103], and are in fact described by a jump-diffusion process.

Anvari *et al.* [98] considered intracranial iEEG time series, which had been recorded during the presurgical evaluation of a subject with drug-resistant focal epilepsy. The multichannel recording [see Fig. 10(a)] lasted for about 2000 s and was taken during the seizure-free interval from within the presumed epileptic focus (seizure-generating brain area), as well as from distant brain regions. Thus, the analyzed data did not have a seizure event. Instead, it contained background iEEG time series. Anvari *et al.* [98] showed that the $D^{(4)}$ coefficient of both time series do not vanish, and modeled the data with a jump-diffusion process. Figure 11 presents the computed diffusion coefficients $D^{(2)}(x)$, jump amplitudes $\sigma^2_\xi(x)$, and jump rates $\lambda(x)$, as well as the respective probability distribution functions, estimated from normalized iEEG time series that contained 4×10^5 data points, for one epilepsy patient.

Carrying out extensive analyses of multi-day, multichannel iEEG recordings from ten epilepsy patients, Anvari *et al.* [98] demonstrated that the dynamics of the epileptic focus is characterized by a stochastic process with a mean diffusion coefficient and a mean jump amplitude that are smaller than those that characterize the dynamics of distant brain regions. Therefore, higher-order Kramers-Moyal coefficients provide extra and highly valuable information for diagnostic purposes.

Note, however, that as a result of the jump processes, estimating the Kramers-Moyal coefficients by Eq. (16) encounters some fundamental drawbacks that have recently been studied [104–106]. Therefore, data-driven reconstruction of the governing equations based on Kramers-Moyal expansion is still an evolving approach, and as it is developed further, it will also find a wider range of applications.

C. Data-driven approach for Mori-Zwanzig projection operator formulation

Mori [107] and Zwanzig [108] developed a formalism that provides a mathematically exact procedure for developing reduced-order models for high-dimensional dynamical systems, such as turbulent flow, as well as data, which are constructed based on projection operators. The essence of the method is reformulating a set of ordinary differential equations (ODEs) into a reduced system for the resolved variables x_r, but still retaining the dynamics of the original system, which implies correctly representing the contribution of the unresolved variable on the resolved physics of the system. The method does so by applying a projection operator to the evolution process of the original dynamic systems described by the set of ODEs, to achieve reduction in their dimensionality.

Mori's formulation leads to a generalized linear Langevin equation, whereas that of Zwanzig produces generalized nonlinear Langevin equation. The equation consists of Markovian, noise, and memory terms, and is an *exact* representation of the dynamics of the model. Thus, the approach may be viewed as a nonlinear generalization of the stochastic Kramers-Moyal expansion, described above. In practice, however, use of the method is computationally difficult, particularly when applied to systems that are described by PDEs; this is discussed below. Comprehensive discussions of the subject are given Mazenko [109], Evans and Morriss [110], and Hijón *et al.* [111].

The approach was originally developed for describing nonequilibrium statistical mechanics of molecular systems, with the goal of solving for the probability density functions and time correlation functions, and was limited to Hamiltonian dynamical systems. Chorin *et al.* [112] extended the formulation to general time-dependent systems, such as those in hydrodynamics and reaction-diffusion systems. They developed their framework for optimal prediction, i.e., obtaining the solution of nonlinear time-dependent problems, described by

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}[\mathbf{x}(t)], \tag{23}$$

for which a full-order solution is too difficult computationally and, in addition, the unresolved part of the initial conditions is uncertain. Here, $\mathbf{x}(t)$ is the state of the system at time t, and $\mathbf{f}[\mathbf{x}(t)]$ represents the dynamic constraints that define the equations of motion of the system, such as, for example, the Navier-Stokes equations for hydrodynamics of Newtonian fluids, which can be generalized to include parametrization

and forcing. Thus, the goal is to determine $\mathbf{f}[\mathbf{x}(t)]$ from the data.

We describe the Mori-Zwanzig formulation by closely following Falkena *et al.* [113]. Consider nonlinear dynamical systems described by Eq. (23), with the initial condition corresponding to a trajectory $x(t)$, $x(t=0)=y$, and an observable $u(y,t)=g[x(t)]$ along the solution of the equation, where g is defined on $\mathbb{R}^n$. Thus, one must have

$$\frac{\partial}{\partial t}u(y,t)=\mathcal{L}u(y,t), \tag{24}$$

with $u(y,0)=g(y)$, where $\mathcal{L}$ is the Liouville operator defined by, $\mathcal{L}u=\sum_{i=1}^{n}R_i(y)\partial u(y,t)/\partial y_i$, with y_i being the ith component of y, and $\mathbf{R}=\{R_1,R_2,\cdots R_i\}$ the vector field. The goal for a linear system is to construct a system of equations for a select subset of m resolved variables $x_r\in\mathbb{R}^m$, with the unresolved variables denoted by $x_u\in\mathbb{R}^{n-m}$, such that $\mathbf{x}=(x_r,x_u)$.

To reduce or map the system of n components onto one with m components, one needs a projection operator P, $P: C(\mathbb{R}^n,\mathbb{R}^k)\to C(\mathbb{R}^m,\mathbb{R}^k)$, with k being the dimension of an arbitrary function $f(x_r,x_u)$ to which the projection is applied. One example of such projection operator is the linear one, $[Pf](x_r,x_u)=f(x_r,0)$, i.e., one that sets all the unresolved variables to zero, keeping only the resolved ones. We denote by Q the complement of P, defined by $Q=I-P$, where I is the identity operator.

The solution of the system (24) is $u(y,t)=[e^{t\mathcal{L}}g](y)$, where $e^{t\mathcal{L}}$ is the evolution (also called the Koopman operator [114]; see below), which propagates an observable with $\mathcal{L}$ and $e^{t\mathcal{L}}$ commuting. Thus, if $g=y_i$, then $x_i(y,t)=e^{t\mathcal{L}}y_i$. Equation (24) is rewritten as

$$\frac{\partial}{\partial t}[e^{t\mathcal{L}}g](y)=[e^{t\mathcal{L}}\mathcal{L}g](y)=[e^{t\mathcal{L}}P\mathcal{L}g](y)+[e^{t\mathcal{L}}Q\mathcal{L}g](y). \tag{25}$$

The second term on the right-hand side of Eq. (25) describes the evolution of the unresolved variables. If one invokes the Duhamel-Dyson identity, namely,

$$e^{t(A+B)}=e^{tA}+\int_0^t e^{(t-s)(A+B)}Be^{sA}ds, \tag{26}$$

and takes $A=Q\mathcal{L}$ and $B=P\mathcal{L}$, then one obtains

$$[e^{t\mathcal{L}}Q\mathcal{L}g](y)=[e^{tQ\mathcal{L}}Q\mathcal{L}g](y)+\int_0^t[e^{(t-s)\mathcal{L}}P\mathcal{L}e^{sQ\mathcal{L}}Q\mathcal{L}g](y)ds, \tag{27}$$

and, therefore, Eq. (25) becomes

$$\frac{\partial}{\partial t}[e^{t\mathcal{L}}g](y)=[e^{t\mathcal{L}}P\mathcal{L}g](y)+[e^{tQ\mathcal{L}}Q\mathcal{L}g](y)+\int_0^t[e^{(t-s)\mathcal{L}}P\mathcal{L}e^{sQ\mathcal{L}}Q\mathcal{L}g](y)ds. \tag{28}$$

In particular, if $g(y)=y_i$, we have $[e^{t\mathcal{L}}g](y)=x_i(y,t)$, and obtain the generalized Langevin equation,

$$\frac{\partial}{\partial t}x_i(y,t)=M_i[x_r(y,t),0]+\mathcal{N}_i(y,t)+\int_0^t K_i[x_r(y,t-s),s]ds, \tag{29}$$

where $\mathcal{N}_i=[e^{tQ\mathcal{L}}Q\mathcal{L}g](y)$ and $K_i=[P\mathcal{L}\mathcal{N}_i](y,t)$. As mentioned above, the Mori-Zwanzig formulation produces a generalized Langevin equation with three terms, namely, the Markov, noise, and memory functions represented, respectively, by M_i, $\mathcal{N}_i$, and the integral on the right-hand side of Eq. (29). The noise term, produced by the uncertainty in the initial conditions, is the solution of the following orthogonal dynamic equation,

$$\frac{\partial}{\partial t}\mathcal{N}_i(y,t)=Q\mathcal{L}\mathcal{N}_i(y,t), \tag{30}$$

with the initial condition, $\mathcal{N}_i(y,0)=Q\mathcal{L}y_i$. It is called orthogonal dynamics because its solution lies in the orthogonal space of projection operator P at all times.

Equation (29) is exact, but determining its solution is not necessarily simpler than the original equation, Eq. (23). The main bottleneck for using the Mori-Zwanzig formulation and utilizing Eq. (29) for constructing dynamical equations for a set of data is determining the numerical solution of Eq. (30), which is difficult. For example, directly evaluating the integral requires storing the solutions from all the previous steps at every time step, which is a difficult task. The "ease" of obtaining the solution depends crucially on the choice of projection operator P, which plays an important role in determining the form and complexity of the orthogonal dynamics equation, Eq. (30). P should be selected such that the orthogonal dynamics system is stable, implying that one must not only retain stabilizing factors in the unresolved dynamics, but also select P such that solving Eqs. (29) and (30) is less complex than solving the original system described by Eq. (23).

A simple example [113,115] illustrates how the method works. Consider the following system of ODEs,

$$\frac{d}{dt}\begin{pmatrix}x_r\\x_u\end{pmatrix}=\begin{pmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{pmatrix}, \tag{31}$$

with the initial condition, $x(0)=(y_r,y_u)$. We wish to derive an equation for x_r only, which is accomplished by solving the equation for x_u by the method of variation of parameters and substituting the result into the equation for x_r to obtain

$$\frac{dx_r}{dt}=a_{11}x_r(t)+a_{12}e^{a_{22}t}y_u+\int_0^t a_{12}e^{a_{22}(t-s)}a_{21}x_r(s)ds. \tag{32}$$

Equation (32), which is exact, exhibits the same behavior as the original system, and the effect of the unresolved variables appears only as the initial condition y_u.

Since the main obstacle to using the Mori-Zwanzig approach is having the right projection operator P, it may be useful to discuss the issue further, to provide some guidance for selecting the operator. As already discussed, Mori's formulation leads to a generalized linear Langevin equation, whereas that of Zwanzig produces a generalized nonlinear Langevin equation. In the former case, the projection operator relies on the inner product defined by, $\langle f,g\rangle=\int f(\mathbf{x})g(\mathbf{x})d\mu(\mathbf{x})$, where $\mu(\mathbf{x})$ is the probability distribution function. Given the inner product, Mori's projection operator is defined onto the span of a set of linearly independent basis

functions $b_i(\mathbf{x})$, so that [116]

$$[Pf]\mathbf{b}(\mathbf{x}) = \sum_i \sum_j \langle f, b_i\rangle [C^{-1}]_{i,j} b_j(\mathbf{x}), \tag{33}$$

with C being the covariance matrix, $C_{ij} = \langle b_i, b_j\rangle$. If the basis functions are orthonormal, then $C = \mathbf{I}$, with $\mathbf{I}$ being the identity matrix, and the projection operator is greatly simplified:

$$[Pf]\mathbf{b}(\mathbf{x}) = \sum_o \langle f, b_i\rangle b_i(\mathbf{x}). \tag{34}$$

However, in Zwanzig's formulation, the observables are a subset of the resolved variables $\mathbf{x}_r$, and the projection operator is defined by direct marginalization of the unresolved variables. If the probability distribution $\mu(\mathbf{x})$ for variable $\mathbf{x}$ is written for resolved and unresolved variables as a density function $\rho(\mathbf{x}_r, \mathbf{x}_u)$ [116], then

$$[Pf](\mathbf{x}_r) = \frac{\int f(\mathbf{x}_r, \mathbf{x}_u)\rho(\mathbf{x}_r, \mathbf{x}_u)d\mathbf{x}_u}{\int \rho(\mathbf{x}_r, \mathbf{x}_u)d\mathbf{x}_u}, \tag{35}$$

which yields a nonlinear function that has been used [117] for developing models of turbulence based on the Mori-Zwanzig formulation.

Alternative ways of getting around the difficulty of selecting the projection operator have also been suggested. For example, Gouasmi *et al.* [115] proposed to approximate the equation for orthogonal dynamics by a less complex one using pseudo-orthogonal dynamics approximation. In their method the memory kernel in the above integral is estimated *a priori* by utilizing full-order solution snapshots. Thus, a pseudo-orthogonal dynamics equation is solved that has the Liouville form, instead of solving the original one. The method is based on the assumption that, for one observable, the semigroup (algebraic structures that consist of a set together with an associative internal binary operation on it) of the orthogonal dynamics operator is a composition operator, akin to semigroups of Liouville operators, hence mimicking their behavior.

Despite the difficulty in developing the right projection operator P and obtaining the numerical solution for the dynamics of the system that is less expensive than solving the original system, the Mori-Zwanzig approach is gradually gaining more recognition and use as a way of discovering reduced-order governing equations for systems for which a considerable amount of data is available.

1. Example 1: Heat conduction in a nanosize system

Chu and Li [118] used the Mori-Zwanzig procedure to derive an equation that describes heat conduction in nanomechanical systems, since the conventional heat conduction equation breaks down at such length scales. They considered a 1D isolated chain of N atoms, divided evenly into n blocks, each of which contained "atoms" with known equilibrium spacing between neighboring atoms, and calculated energy transport between the blocks. Thus, the local energy density was selected as the coarse-grained variable, for which a generalized Langevin equation was derived using the Mori-Zwanzig procedure. The propagating operator $\mathcal{L}$ was defined by

$$\mathcal{L} \equiv v_0 \frac{\partial}{\partial x_0} + \frac{f(x_0)}{m}\frac{\partial}{\partial v_0}, \tag{36}$$

where x_0 and v_0 are, respectively, the initial position and velocity of the molecules, m is their mass, and $f(x)$ is the force, i.e., $f = -\nabla E(x)$, with E being the potential energy. They showed that the calculated results using the Mori-Zwanzig method agrees with the results obtained with nonequilibrium molecular dynamics simulations in which they imposed a temperature gradient between the two ends of the chain of 250 atoms.

2. Example 2: Reduced-order equation for turbulent flow

Tian *et al.* [116] used extensive data for isotropic turbulence to drive the projection operator of Mori-Zwanzig approach and construct a reduced-order Navier-Stokes equation. If the equation is spatially discretized, then one obtains the following set of nonlinear equations for the fluid's velocity $\mathbf{v}(t)$:

$$\frac{d\mathbf{v}(t)}{dt} = R[\mathbf{v}(t)], \tag{37}$$

which is similar to Eq. (23), where R is the nonlinear function that represents the spatially discretized right-hand side of the Navier-Stokes equations. Computations for fully resolved dynamics of the Navier-Stokes equations is prohibitive for any physical problem. Thus, to develop a reduced-order model for turbulence, the velocity field is usually coarse-grained using a spatial filter, which reduces the range of scales that must be resolved. Suppose that $\bar{\mathbf{v}}(t)$ is the filtered fluid velocity. Then, as described above, according to the Mori-Zwanzig formulation, Eq. (29) for $\bar{\mathbf{v}}(t)$, in vector form, is given by

$$\frac{d\bar{\mathbf{v}}(t)}{dt} = M[\bar{\mathbf{v}}(t)] + \mathcal{N}(t) - \int_0^t K[\bar{\mathbf{v}}(t-s), s]ds, \tag{38}$$

which is the nonlinear version of the formulation. In the linear formulation, i.e., in terms of Mori's original derivation, the generalized Langevin equation for the linearly independent basis functions $\mathbf{b}(t)$ is given by

$$\frac{d\mathbf{b}(t)}{dt} = \mathbf{M}\cdot\mathbf{b}(t) + \mathcal{N}(t) - \int_0^t \mathbf{K}(t-s)\cdot\mathbf{b}(s)ds. \tag{39}$$

The advantage of Mori's projection operator is that, due to the linearity of the projected low-dimensional functions, the derivation of the kernel $\mathbf{K}$ is significantly simpler.

Extensive data were obtained by numerical simulation of fully resolved discrete Eulerian Navior-Stokes equations, given by

$$\frac{\partial v_i}{\partial t} + \frac{\partial v_i v_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \nu\frac{\partial^2 v_i}{\partial x_j^2}, \tag{40}$$

where ν is the kinematic viscosity, and p is the pressure that was computed by solving the Poisson's equation for p. The data were used to extract the kernel and the noise term in Eq. (39) by computing a two-point correlation function and relating them to each other by an iterative process [119].

Figure 12 compares the Frobenius norm of the memory kernel (normalized by its corresponding Markov operator) for

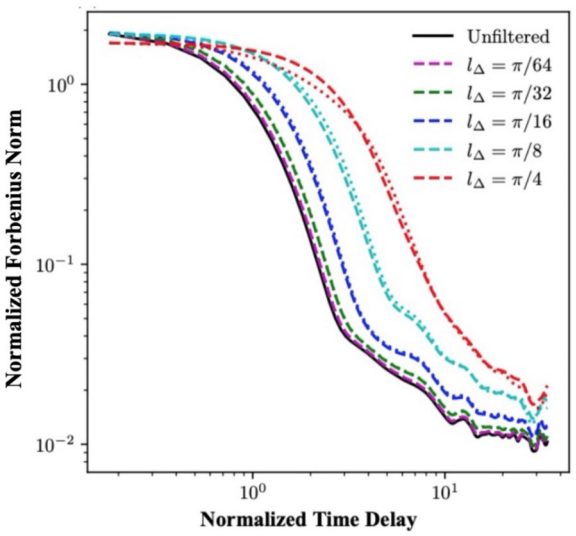

FIG. 12. Normalized Frobenius norm of the learned memory kernel for observable set as a function of normalized time delay. Two types of spatial filters, Gaussian and box filters, with various filtering length scales l_Δ, were applied to the variables of the physical space. Based on Ref. [116].

a set of observable in the original data with the results obtained with a Gaussian filter of various resolution, as measured by filtering length l_Δ. The Frobenius norm of an $m \times n$ matrix is defined as the square root of the sum of the squares of its elements. As discussed by Tian *et al.* [116], the Frobenius norm of the memory kernel does not vanish with a finite time delay, but becomes two to three orders of magnitude smaller at a time delay around several Kolmogorov timescales (i.e., the smallest timescale in turbulent flow), hence indicating that using finite support in the memory integral can be a reasonable assumption, because the contributions from large time delays are generally negligible. Moreover, the effect of the filtering length scale l_Δ is significant. With larger l_Δ the temporal decay of the memory kernel becomes slower, making the finite memory length longer, hence indicating a shift of dynamical contributions from the Markov term to memory integral.

As mentioned above, even though the Mori-Zwanzig formalism was developed over 50 years ago, due to the intensive computations that are required for determining the kernel in the integro-differential equation that represents the generalized Lagevin equation, as well as the complexity of selecting the projection operator, only very recently have the applications of the method begun to emerge. They include developing a reduced-order model for turbulence by Parish and Duraisamy [120,121] and Maeyama and Watanabe [122]. Li and Stinis [123] developed a reduced-order model for uncertainty quantification, while Stinis [124] presented a series of higher-order models for the Euler equation based on the Mori-Zwanzig formulation. The research field is finally emerging.

D. Koopman and Perron-Frobenius transfer operators

When the available data are for a system that exhibits multiscale features, reconstructing the governing equation is a very difficult task as it involves prohibitive computations. An example is a dynamical system that consists of many subsystems operating over distinct times scales, such that some of the subsystems act over short timescales, whereas others respond slowly to any external driver. Alternatively, the system may possess intrinsic properties that act over distinct timescales, such as fast vibrations and slow conformational changes of molecules in a molecular system. Such systems suffer from what Richard Ernest Bellman, the American applied mathematician, called [125] the *curse of dimensionality*, which refers to phenomena that arise when one analyzes and organizes data in a high-dimensional space, not in a low-dimensional one, such as the 4D space of everyday life. In such cases one resorts to a data-driven dimensionality reduction technique, examples of which were already described above, for the dynamical system. In other words, although the true governing equation may be a function of multiple variables in the form of a PDE, one tries to represent and reproduce the important features of the system and the data by a much simpler equation.

There are a variety of such techniques, some of which were already described, and other are described in this and the following subsections. Here, we describe those that are based on the transfer operator theory. Such methods involve approximating the operators and their eigenvalues, eigenfunctions, and eigenmodes, and have been used to analyze complex problems in molecular dynamics, fluid flow, applied physics, and various engineering disciplines. A good review of the subject is given by Klus *et al.* [126], on which we partly rely for our discussions.

First, let us define a transfer operator. Suppose that $\{\mathbf{X}_t\}$ (with $t \geqslant 0$) is an *autonomous* or *time-homogeneous* stochastic process, i.e., one for which the distribution of $\mathbf{X}_t$, conditioned to $\mathbf{X}_s = s$, depends only on x and $(t - s)$ with $t \geqslant s \geqslant 0$. Then, the transition density function, $p_\tau : \mathbb{X} \times \mathbb{X} \to [0, \infty]$ of the process $\mathbf{X}_t$ is defined by

$$\mathbb{P}[\mathbf{X}_{t+\tau} \in \mathbb{A} | \mathbf{X}_t = x] = \int_{\mathbb{A}} p_\tau(x, y) dy, \tag{41}$$

where $\mathbb{A}$ is a measurable set (dataset). Clearly, $\mathbb{P}$ is a conditional probability, implying that $p_\tau(x, y)$ is the conditional probability density of $\mathbf{X}_{t+\tau} = y$, given that $\mathbf{X}_t = x$. Then, if $f_t(y) \in L^\infty(\mathbb{X})$ is an observable of, or set of data for, the system, the *Koopman operator* $\mathcal{K}_\tau : L^\infty(\mathbb{X}) \to L^\infty(\mathbb{X})$ is defined [114,127] by (for a recent review of Koopman operator dynamical models see Bevanda *et al.* [128])

$$\mathcal{K}_\tau f_t(x) = \int_{\mathbb{X}} p_\tau(x, y) f_t(y) dy, \tag{42}$$

where τ is a given lag time. Equation (42) clearly indicates that the Koopman operator, which is infinite-dimensional and linear, describes the evolution of the observable (or the dataset). Another operator is the *Perron-Frobenius* operator (propagator), $\mathcal{P}_\tau : L^1(\mathbb{X}) \to L^1(\mathbb{X})$, defined by

$$\mathcal{P}_\tau p_t(x) = \int_{\mathbb{X}} p_\tau(y, x) p_t(y) dy, \tag{43}$$

where $p_t \in L^1(\mathbb{X})$ is the probability density of the system. Note that $\mathcal{P}_\tau$, an infinite-dimensional operator, describes the evolution of densities. Both $\mathcal{K}_\tau$ and $\mathcal{P}_\tau$ are adjoint of each other, when considered with respect to the duality pairing

(inner product) defined by $\langle f, g\rangle = \int_{\mathbb{X}} f(x)g(x)dx$. The assumption that the dynamical process $\mathbf{X}_t$ is homogeneous implies that, $\mathcal{P}_{\sigma+\tau} = \mathcal{P}_\sigma \mathcal{P}_\tau$, and similarly for the Koopman operator.

For many applications, one is interested in the system's evolution with respect to the equilibrium density. Hence, consider a density π. It is called *invariant* or *equilibrium* density, if $\mathcal{P}_\tau \pi = \pi$, implying that π is an eigenfunction of $\mathcal{P}_\tau$ with the corresponding eigenvalue being one. If $L^1_\pi(\mathbb{X}) \ni u_t(x) = \pi(x)^{-1} p_t(x)$ is a probability density with respect to the equilibrium density π, then the Perron-Frobenius operator *with respect to the equilibrium density* is defined by

$$\mathcal{T}_\tau u_t(x) = \int_{\mathbb{X}} \frac{\pi(y)}{\pi(x)} p_\tau(y, x) u_t(y) dy. \tag{44}$$

As usual, one is interested in the eigenvalues $\lambda_i(\tau) \in \mathbb{C}$ and eigenfunctions $\phi_i : \mathbb{X} \to \mathbb{C}$ of such operators, defined by the standard way, namely, $\mathcal{A}_\tau \phi_i = \lambda_i(\tau)\phi_i$, where $\mathcal{A}_\tau$ is any of the three operators defined above. As an example, consider a 1D Ornstein-Uhlenbeck process, given by [see also Eq. (20)]

$$d\mathbf{X}_t = -\beta D \mathbf{X}_t dt + \sqrt{2D} d\mathbf{W}_t, \tag{45}$$

with $\mathbf{X}_t$ being a 1D Brownian motion (i.e., the Wiener process), and β and D the friction and diffusion coefficients. Then, the eigenvalues and eigenfunctions of the Koopman operator associated with the Ornstein-Uhlenbeck process are given by

$$\lambda_i = \exp[-\beta D(i-1)\tau] \tag{46}$$

and

$$\phi_i(x) = \frac{1}{\sqrt{(i-1)!}} H_{i-1}(\sqrt{\beta}\, x), \tag{47}$$

with H_i being the ith probability's Hermite polynomial [129], and $i = 1, 2, \cdots$.

Our main focus is the application of the Koopman operator, whose eigenvalues and eigenfunctions, together with the *Koopman modes* that are vectors, allow one to reconstruct a dynamical system and propagate its state [130]. In essence, the eigenfunctions of the Koopman operator provide intrinsic coordinates that globally linearize the original nonlinear dynamics. An important point to remember is that the Koopman eigenvalues and eigenfunctions are *independent* of observation state, implying that data acquired by various sensors would yield the same results for the eigenvalues and eigenfunctions, if the information that the data contain is sufficiently rich.

Suppose that each component g_j of the full-vector observables (data points), $g_j(x) = x_j$ ($j = 1, \cdots, d$, with d being the dimension of the data vector), is written as, $g_j(x) = \sum_i \phi_i(x)\eta_{ij}$. Here, η_i are the components of the Koopman modes defined by, $\eta_i = [\eta_{1i}, \eta_{2i}, \cdots, \eta_{di}]^{\mathrm{T}}$. Therefore, we obtain $g(x) = x = \sum_i \phi_i(x)\eta_i$, and

$$\begin{aligned}\mathcal{K}_\tau g(x) &= \mathbb{E}[g(\mathbf{X}_\tau)|\mathbf{X}_0 = x] = \mathbb{E}[\mathbf{X}_\tau|\mathbf{X}_0 = x] \\ &= \sum_i \lambda_i(\tau)\phi_i(x)\eta_i.\end{aligned} \tag{48}$$

The key to data-driven dimensionality reduction and reconstruction of the dynamical equation is the recognition that since the operator is infinite-dimensional, computing its eigenvalues, eigenfunctions, and eigenmodes numerically entails projecting the operator onto a finite-dimensional space, spanned by a given set of basis functions, which we describe next, to make the calculations manageable.

Consider a pair of data vectors, x_i and y_i, with $x_i = \mathbf{X}_{t_i}$ and $y_i = \mathbf{X}_{t_i+\tau}$, assuming that we do not necessarily know the system's underlying dynamics. Writing, $\mathbf{X} = [x_1, x_2, \cdots, x_m]$ and $\mathbf{Y} = [y_1, y_2, \cdots, y_m]$, with m being the number of measurement points, and assuming that a long trajectory, $z = [z_0, z_1, \cdots]$ of the system is given, with $z_i = \mathbf{X}_{t_0+hi}$, where h is the time step size, we obtain, $\mathbf{X} = [z_0, z_1, \cdots, z_{m-1}]$ and $\mathbf{Y} = [z_{n_\tau}, z_{n_\tau+1}, \cdots, z_{n_\tau+m-1}]$, i.e., $\mathbf{Y}$ is obtained by shifting $\mathbf{X}$ by the lag time τ, where $\tau = n_\tau h$. A similar method is used if more than one trajectory is given, in which case one has data matrices, rather than vectors.

For some methods of projecting an infinite-dimensional operator onto a finite-dimensional one, one needs a set of uniformly bounded basis functions or observables (sometimes they are called *dictionary*), $[\psi_1, \psi_2, \cdots, \psi_k] \subset L^\infty(\mathbb{X})$, to represent the eigenfunctions, which could be any type of function. The crucial point is to select an optimal size of the set of such basis functions. A limited set with a few basis functions may not accurately represent the eigenfunctions, whereas if the set is too large, one may encounter overfitting [127]. There are several methods for data-driven approximation of transfer operators, particularly the Koopman operator, some of which were described by Klus *et al.* [126]. In what follows we describe such methods.

1. Time-lagged independent component analysis (TICA)

The method was proposed by Molgedey and Schuster [131] and developed further by Hyvärinen *et al.* [132]. It has been used in molecular dynamics as a preprocessing step to lower the size of the state space, which is accomplished by projecting the dynamics onto the main coordinates [133,134]. For the method to be applicable, the time-lagged independent components must be uncorrelated and maximize the autocovariances at lag time τ [132,133]. If the system is reversible, which would be the case if detailed balance holds, i.e., for all $x, y \in \mathbb{X}$ one has $\pi(x)p_\tau(x, y) = \pi(y)p_\tau(y, x)$, then the TICA coordinates are the eigenfunctions of $\mathcal{K}_\tau$ (or of $\mathcal{T}_\tau$), projected onto the space that is spanned by the linear basis functions, $\psi(x) = x$. If we define a time-lagged covariance matrix $C_{ij}(\tau) = \langle \mathbf{X}_{t,i}\mathbf{X}_{t+\tau,j}\rangle_t$, then given the data vectors $\mathbf{X}$ and $\mathbf{Y}$, the estimators C_0 and C_τ for the true covariance matrices $\mathbf{C}(0)$ and $\mathbf{C}(\tau)$ are computed by

$$C_0 = \frac{1}{m-1}\sum_{i=1}^m x_i x_i^{\mathrm{T}} = \frac{1}{m-1}\mathbf{X}\mathbf{X}^{\mathrm{T}}, \tag{49}$$

$$C_\tau = \frac{1}{m-1}\sum_{i=1}^m x_i y_i^{\mathrm{T}} = \frac{1}{m-1}\mathbf{X}\mathbf{Y}^{\mathrm{T}}. \tag{50}$$

Note that the time-lagged independent components are the solution of the eigenvalue problem, $C_\tau \xi_i = \lambda_i C_0 \xi_i$. A modification of TICA, called temporal decorrelation source separation (TDSEP), proposed by Ziehe and Müller [135], utilizes several time-lagged correlation matrices, rather than

one, and has been used heavily by those who utilize machine-learning techniques.

2. Dynamic mode decomposition (DMD)

Schmid [136] proposed the DMD method for time series in fluid mechanical systems, to identify their coherent structures. The algorithm, an effective method for capturing the essential features of numerical or experimental data for a flow field, computes a set of modes, each of which is associated with a fixed oscillation frequency and decay and growth rate, and represents approximation of the modes and eigenvalues of the Koopman operator [127,128] for the basis function $\psi(x) = x$. Several extensions of the method have also been proposed [137–140], while Mezić [141] provides a good review of the applications of Koopman operator to fluid systems, where references to his earlier pioneering work are also given. In particular, Jovanović *et al.* [140] developed a sparsity-promoting variant of the original DMD algorithm in which sparsity was induced by regularizing the least-squared differences between the matrix of snapshots of a system and a linear combination of the modes, with an additional term that penalizes the L^1-norm—the sum of the magnitudes of the vectors in a space—of the vector of the DMD amplitudes. As the name suggests, the only assumption of the algorithm about the structure of the model is that, there are only a few important terms that govern the dynamics of a system, implying that the searched-for equations are sparse in the space of possible functions, an assumption that holds for many physical systems.

Given the data vectors $\mathbf{X}$ and $\mathbf{Y}$ described above, the main idea in the DMD algorithm is that there exists a linear operator $\mathcal{M}_{\rm DMD}$, defined by, $y_i = \mathcal{M}_{\rm DMD} x_i$, which, due to the nonlinearity of the system's dynamics, cannot be solved exactly. Thus, one computes $\mathcal{M}_{\rm DMD}$ in such a way as to minimize the norm $\|\mathbf{Y} - \mathcal{M}_{\rm DMD}\mathbf{X}\|$ whose solution is

$$\mathcal{M}_{\rm DMD} = \mathbf{Y}\mathbf{X}^{+} = (\mathbf{Y}\mathbf{X}^{\rm T})(\mathbf{X}\mathbf{X}^{\rm T})^{+}. \tag{51}$$

The eigenvalues and eigenvectors of $\mathcal{M}_{\rm DMD}$—the DMD eigenvalues and modes—are the solutions of

$$\mathcal{M}_{\rm DMD}\xi_i = \lambda_i \xi_i. \tag{52}$$

Equations (51) and (52) indicate clearly that there is a close relationship between DMD and TICA algorithms. The modes of the former are the right eigenvectors of $\mathcal{M}_{\rm DMD}$, whereas the TICA coordinates are defined to be the right eigenvectors of the transposed TICA matrix, implying that the TICA coordinates are the left eigenvectors of the DMD matrix, while the DMD modes are the left eigenvectors of the TICA matrix.

3. Variational approach of conformation dynamics (VAC)

This method, developed by Noé and Nüske [142] and Nüske *et al.* [143,144], is applicable to only reversible systems (see above), and allows use of arbitrary basis functions. It computes the eigenfunctions of the Koopman operator and utilizes the transformed data matrices, $\mathbf{\Psi}_X = [\psi(x_1), \psi(x_2), \cdots, \psi(x_m)]$ and $\mathbf{\Psi}_Y = [\psi(y_1), \psi(y_2), \cdots, \psi(y_m)]$, to compute the covariance matrices C_0 and C_τ defined by Eqs. (49) and (50). They are given by

$$C_0 = \frac{1}{m-1}\sum_{i=1}^{m} \psi(x_i)\psi(x_i)^{\rm T}, \tag{53}$$

$$C_\tau = \frac{1}{m-1}\sum_{i=1}^{m} \psi(x_i)\psi(y_i)^{\rm T}, \tag{54}$$

which are completely similar to Eqs. (49) and (50). Let, $\mathcal{M}_{VAC} = C_0^{+} C_\tau$, where C_0^{+} is the Moore-Penrose pseudo-inverse of C_0 [145]. Then, $\mathcal{M}_{VAC}$ represents a finite-dimensional approximation of the Koopman operator $\mathcal{K}_\tau$ (or of $\mathcal{T}_\tau$), and the eigenfunctions of the operator are approximated by the eigenvectors of $\mathcal{M}_{\rm VAC}$.

4. Extended dynamic mode decomposition (EDMD)

This method, developed by Williams *et al.* [130,146], is used to compute finite-dimensional approximations of the Koopman operator and its eigenvalues, eigenvectors, and eigenmodes. One minimizes $\|\mathbf{\Psi}_Y - \mathcal{M}_{\rm EDMD}\|$, whose solution is

$$\mathcal{M}_{\rm EDMD} = \mathbf{\Psi}_X \mathbf{\Psi}_Y^{+} = \mathbf{C}_\tau^{\rm T}\mathbf{C_0}^{\rm T}. \tag{55}$$

Thus, what EDMD does is that, instead of assuming a linear relationship between the data $\mathbf{X}$ and $\mathbf{Y}$, it develops a linear relationship between $\mathbf{\Psi}_X$ and $\mathbf{\Psi}_Y$, the transformed data. The eigenfunctions of the Koopman operator are then obtained by

$$\phi_i(x) = \xi_i^{*}\psi(x). \tag{56}$$

Example: Deca polyalanine. Polyalanine is a dipeptide repeat protein that is believed to play an important role in the development of such diseases as amyotrophic lateral sclerosis and frontotemporal dementia [147,148]. Its properties, such as its aggregation and folding, have been studied extensively by molecular dynamics (MD) simulations [149,150]. Nüske *et al.* [143,144] studied deca alanine, a short version of the protein by equilibrium MD simulations using Amber03 force field, which was then reanalyzed by Klus *et al.* [126]. An important set of parameters for the analysis is the so-called *implied timescales* t_i that are independent of the lag time τ. If, however, we define and estimate t_i by

$$t_i = -\frac{\tau}{\log|\lambda_i|}, \quad i = 2, 3, \cdots, \tag{57}$$

where λ_i is the ith eigenvalue computed by the EDMD/VAC approaches, then the error will decrease as a function of the time lag [151]. The analysis is carried out in three steps:

(1) One extracts a set of internal molecular coordinates from the simulation data, and applies the TICA algorithm to it (see above). Klus *et al.* [126] selected all 16 diehedral angles on the protein's backbone as the internal molecular coodinates. Figure 13(a) presents the first five t_i as a function of the time lag τ, estimated by the TICA.

(2) The data are projected onto the leading M eigenvectors of TICA, thereby performing the first dimensionality reduction. M is the smallest number such that [152] the sum of the first M squared eigenvectors is larger than 95 percent of the total sum of the squared eigenvectors. Figure 13(b) shows how M is selected as a function of τ.

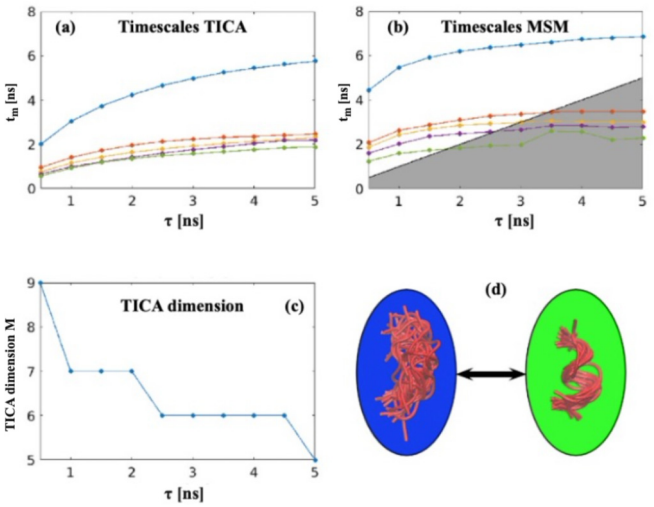

FIG. 13. The extended dynamic mode decomposition (EDMD) workflow in molecular dynamics of deca alanine. (a) Leading implied timescales t_m (in nanoseconds) as estimated by time-lagged independent component analysis (TICA) as a function of the lag time. (b) Effective dimension M, selected such that the sum of the first M squared eigenvectors is larger than 95% of the total sum of the squared eigenvectors. (c) Leading implied timescales t_m estimated by a Markov state model after projecting the data onto the first M TICA eigenvectors and discretizing the data set into 50 states using k-means. (d) Visualization of effective coarse grained dynamics. All the Markov state model states are assigned to two macrostates. Overlay of representative structures from both macrostates shows that the dynamics between them corresponds to helix formation. Macrostates are drawn proportionally to their stationary probability. Based on Ref. [126].

(3) A clustering method is used to discretize the reduced data. Klus *et al.* [126] used the k-means clustering method [153] to partition the data into 50 discrete states, and a Markov state model (solution of the master equation) [154,155] was estimated from the discretized time series. Figure 13(c) shows the first five implied timescales obtained from the Markov state model. This completes the task of developing a converged model.

Once such a model is obtained, the behavior of the system can be analyzed furthur. In this particular example, the implied timescale t_2 turned out to be the slowest one, being larger than the time lag used in the analysis. Thus, Klus *et al.* [126] used an advanced version of the Perron cluster-cluster analysis (PCCA) algorithm [156,157], namely, the PCCA+ algorithm, to coarse grain all the Markov state models into only two macrostates. When randomly selected trajectory frames from the two macrostates were analyzed, it turned out that the slow dynamical process in the data corresponded to the formation of a helix. This is shown in Fig. 13(d).

5. Deep-learning approach (DLA)

This approach was proposed by Lusch *et al.* [158] for extracting the eigenfunctions of the Koopman operator from data; for earlier related works see Refs. [159,160]. According to the universal approximation theorem (UAT) [161,162], a neural network with sufficiently large number of hidden units and a linear output layer (see above) is capable of representing any arbitrary function. The UAT imposes limits on what neural networks can theoretically learn by establishing the density of an algorithmically generated class of functions within a given function space of interest. Such results typically concern the approximation capabilities of the feedforward architecture on the space of continuous functions between two Euclidean spaces, with the approximation being with respect to the compact convergence topology. It is, therefore, natural to use a neural network to represent the Koopman eigenfunctions. In the present context, the goal of the neural network [158] is to identify a few key intrinsic coordinates (see above), $\mathbf{y} = \phi(\mathbf{x})$, spanned by a set of Koopman eigenfunctions $\phi : \mathbb{R}^n \to \mathbb{R}^p$, as well as a dynamical system $\mathbf{y}_{k+1} = \mathbf{K}\mathbf{y}_k$, where $\mathbf{K}$ represents a finite-dimensional approximation of the Koopman operator. As discussed by Lusch *et al.* [158], three types of the loss or cost functions are used in training the neural network, for which there are three requirements for the network, which are as follows:

(1) One tries to identify a few intrinsic coordinates, $\mathbf{y} = \phi(\mathbf{x})$, which are useful for reconstruction, where the dynamics evolve, along with $\mathbf{x} = \phi^{-1}(\mathbf{y})$ to be able to recover state $\mathbf{x}$, which is achieved using an autoencoder, with ϕ being is the encoder and ϕ^{-1} the decoder. The dimension p of the autoencoder subspace is treated as a hyperparameter of the network, whose choice is guided by knowledge of the system. To reconstruct the autoencoder accurately, the loss function used is, $\|\mathbf{x} - \phi^{-1}[\phi(\mathbf{x})]\|$.

(2) To discover the Koopman eigenfunctions, one also requires linear dynamics. The neural network learns the linear dynamics $\mathcal{K}$ on the intrinsic coordinates mentioned above, i.e., $\mathbf{y}_{k+1} = \mathbf{K}\mathbf{y}_k$, for which the loss function used is, $\|\phi(\mathbf{x}_{k+1}) - \mathbf{K}\phi(\mathbf{x}_k)$. If the linear prediction is to be enforced over m time steps, then the loss function used will be $\|\phi(\mathbf{x}_{k+m}) - \mathbf{K}^m\phi(\mathbf{x}_k)\|$

(3) One, of course, requires the intrinsic coordinates to predict the future step(s). To identify linear dynamics in the matrix $\mathbf{K}$, the loss function used for one step is $\|\mathbf{x}_{k+1} - \phi^{-1}[\mathbf{K}\phi(\mathbf{x})]\|$, and $\|\mathbf{x}_{k+m} - \phi^{-1}[\mathbf{K}^m\phi(\mathbf{x}_k)]\|$ over m time steps.

As usual, trajectories are generated from random initial conditions and are used in training the neural network. The trajectories are divided into training, validation, and test sets. Models are trained on the training set and compared on the validation set, which is also used for early stopping to prevent overfitting.

Example: High-dimensional nonlinear fluid flow [158]. Consider nonlinear fluid flow past a circular cylinder at Reynolds number (based on the cylinder's diameter), Re = 100, which is characterized by vortex shedding. Noack *et al.* [163] showed that the high-dimensional dynamics of the flow evolve on a low-dimensional attractor, given by a slow manifold (a topological space that locally resembles Euclidean space near each point) in the following model:

$$\frac{dx_1}{dt} = \mu x_1 - \omega x_2 + A x_1 x_2, \tag{58}$$

$$\frac{dx_2}{dt} = \omega x_1 + \mu x_2 + A x_2 x_3, \tag{59}$$

$$\frac{dx_3}{dt} = -\lambda\left(x_3 - x_1^2 - x_2^2\right), \tag{60}$$

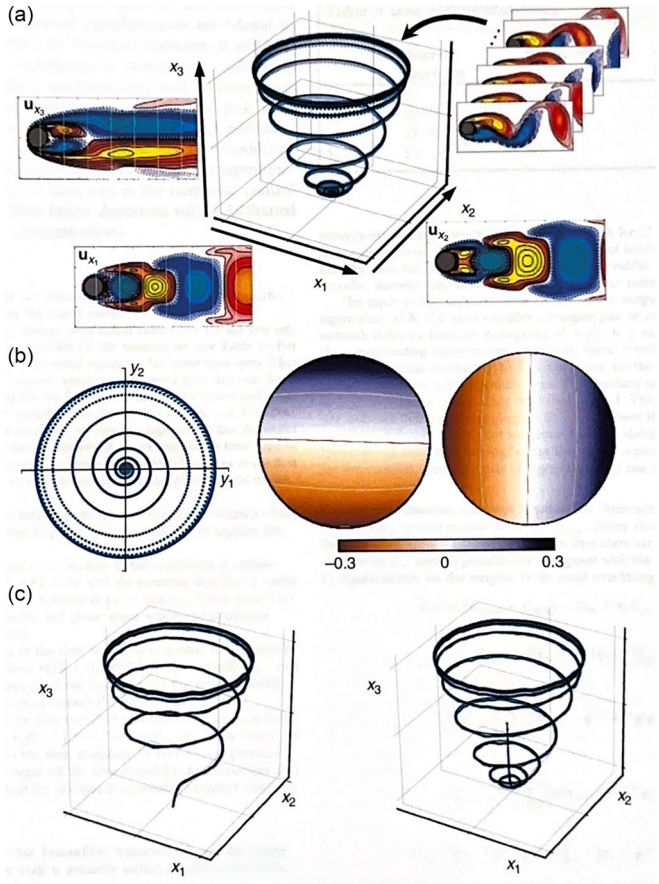

FIG. 14. Learned Koopman eigenfunctions for the model of fluid flow past a circular cylinder. (a) Reconstruction of trajectory by linear Koopman model with two states. Modes for each of the state space variables x are shown along the coordinate axes. (b) Koopman reconstruction in eigenfunction coordinates y, along the eigenfunctions, $y = \phi(x)$. (c) Examples of trajectories that begin off the attractor, which the Koopman model reconstructs, given only the initial condition. Based on Ref. [195].

where μ is the damping rate, and ω is the frequency, with A being a constant. This is a mean-field model with a stable limit cycle that corresponds to von Kármán vortex shedding, and an unstable equilibrium that corresponds to a low-drag fluid system. Lusch *et al.* [158] generated trajectories of this model by solving Eqs. (58)–(60) with $\mu = 0.1$, $\omega = 1$, $A = -0.1$, and $\lambda = 10$ and used them to train the Koopman neural network; see Ref. [158] for the details of the computations. If $R^2 = y_1^2 + y_2^2$, where y_1 and y_2 are the eigenfunction coordinates, then during the training, μ and ω were allowed to vary along level sets of the radius in the eigenfunction coordinates, i.e., both parameters depended on R. The eigenvalues varied continuously, and the computations indicated that ω is very close to its true value of 1, but μ varies significantly. Figure 14 shows a sample of the results.

For application of Koopman operator to the problem of *controlling* nonlinear systems, a difficult and very important problem in industrial applications, see Kaiser *et al.* [164]. Since, as discussed above, the eigenfunctions of Koopman operator provide intrinsic coordinates along which the dynamics behave linearly, Kaiser *et al.* [164] developed a method by which the control is formulated directly in the intrinsic coordinates. The resulting control architecture was dubbed *Koopman reduced-order nonlinear identification and control* (KRONIC). The eigenfunctions were approximated by data-driven regression and power series expansions, based on the partial differential equation that governs the infinitesimal generator of the Koopman operator. Kaiser *et al.* [164] argued that, (a) one must first *validate* the eigenfunctions, (b) lightly damped eigenfunctions can be extracted using the EDMD algorithm (see above), and (c) such eigenfunctions are particularly relevant for control because they correspond to nearly conserved quantities that are associated with persistent dynamics.

6. Spectral and Galerkin methods

Giannakis [165] developed a framework for representing the Koopman and Perron-Frobenius operators by a smooth orthonormal basis in the L^2 space of the dynamical system under study, which was obtained from time-ordered data through the diffusion maps algorithm. A diffusion map [166,167] (Laplace operator), a dimensionality reduction or feature extraction algorithm, computes a family of embeddings of a dataset into a low-dimensional Eulidean space, the coordinates of which are computed based on the eigenvectors and eigenvalues of a diffusion (Laplace) operator acting on the data. We will come back to diffusion maps shortly. Using the representation by the smooth orthonormal basis, Giannakis [165] computed the Koopman eigenfunctions using a regularized convection-diffusion operator (a diffusion operator augmented by a drift term). He then utilized the eigenfunctions for dimension reduction maps for dynamic systems with high smoothness for a given observation state, and considered several types of systems. One was those with pure point spectra, for which he constructed a decomposition of the generator of the Koopman group into mutually commuting vector fields that transform under changes of observation state, reconstructed in the data space representing the pushforward map (a linear approximation of smooth maps) in the Koopman eigenfunction basis. The second type of systems studied by Giannakis [165] was those with a noisy time series, i.e., one in which each data point x_i is given by, $x_i = \tilde{x}_i + \zeta_i$, where ζ_i are independent and identically distributed random variable with zero mean, and finite second through fourth moments, and $\tilde{x}_i$ is the noiseless data point.

As discussed by Giannakis [165], the advantages of his method are fourfold: (a) one is able to construct nonlinear dimension reduction maps based on Koopman eigenfunctions with projectible dynamics and small roughness on the data manifold (a manifold is a topological space that locally resembles Euclidean space near each point); (b) one can decompose the dynamical vector field into a sum of mutually commuting vector fields, which are reconstructed in the data space by a spectral representation of the pushforward map for vector fields on manifolds; (c) the method improves the efficiency and noise robustness of the approaches for the Koopman eigenvalue problem through delay-coordinate maps; and (d) the algorithm predicts the dynamic evolution of arbitrary probability densities and the expectation values of the observables. For the related work in which the orthonormal set is constructed through eigenfunctions of the Laplace-Beltrami

operator, and computed via sparse graph-theoretic algorithms, see Giannakis and Majda [168]. The Laplace-Beltrami operator is a generalization of the usual Laplace operator for functions that are defined on a submanifold in Euclidean space.

Let us point out that, studying the transport and mixing properties of flows in a variety of systems, Froyland and Padberg [169] succeeded in connecting the classical geometrical approach via invariant manifolds with a probabilistic approach via a transfer operator, which in their study was a Perron-Frobenius operator described above. They demonstrated that for nondivergent fluid flow, the eigenvectors of the transfer operator efficiently decomposes the domain into invariant regions. If the flow is dissipative and chaotic, however, then a decomposition into invariant regions is nonexistent, but the Perron-Frobenius transfer operator identifies almost-invariant sets. They also demonstrated that, for a mixing, periodically driven fluid flow, the sets bounded by stable and unstable manifolds are almost invariant, and that the transfer operator can identify such sets.

E. Diffusion maps

We already defined diffusion maps, and in this section we provide the details of the algorithm for dimensionality reduction. A good review and description of the method is given by Trstanova *et al.* [170]. A diffusion process is modeled by a random walk between nearest-neighbor points [171], and the method of diffusion maps exploits this by observing that a step of the walk between two close data points is more likely than one between two widely separated ones. As usual, suppose that X is a set of data points, while $\mathcal{D}$ is their distribution in X. We define the connectivity c between two data points x and y as the probability of taking one step of the walk from x to y, which is usually specified in terms of a kernel function of the two points: $c : X \times X \to \mathbb{R}$. One well-known example of $c(x, y)$ is a Gaussian distribution. Note that $c(x, y) = c(y, x)$, and that the choice of $c(x, y)$ depends on the application. One then constructs a *normalized graph Laplacian*, which is a reversible discrete-time Markov chain on X, given by

$$d(x) = \int_X c(x, y) d\mathcal{D}(y), \tag{61}$$

and then one defines a new normalized kernel by, $k(x, y) = c(x, y)/d(x)$, so that $\int k(x, y) d\mathcal{D}(y) = 1$. $k(x, y)$ is the transition probability of one step from x to y.

In the next step, one constructs a transition matrix $\mathbf{M}$ for the random walk (Markov chain) on the the data X. Then, $\mathbf{M}^t$ is the transition matrix for a t-step walk on X. Defining a diffusion matrix $\mathbf{L}$ whose entries are, $L_{ij} = c(x_i, x_j)$, a new kernel is introduced through

$$L_{ij}^{(\alpha)} = \frac{L_{ij}}{[d(x_i) d(x_j)]^{\alpha}}. \tag{62}$$

The parametrization by α is necessary for tuning the influence of the data point density on the infinitesimal transition of diffusion. $\alpha = 0$ reduces the algorithm to the classical graph Laplacian normalization, representing maximal influence of sampling density. $\alpha = 1/2$ is used when one is interested in describing the long-time behavior of the point distribution of a system of stochastic differential equations, and the resulting Markov chain approximates the Fokker-Planck equation, whereas if the sampling of the data is not related to the geometry of the manifold that one is interested in describing, one sets $\alpha = 1$, indicating no influence of sampling density, and the diffusion operator approximates the aforementioned Laplace-Beltrami operator. If $\mathbf{D}$ and $\mathbf{D}^{(\alpha)}$ are diagonal matrices whose entries are given by $D_{ii} = \sum_j L_{ij}$ and $D_{ii}^{(\alpha)} = \sum_j L_{ij}^{(\alpha)}$, then $\mathbf{L}^{(\alpha)} = \mathbf{D}^{-(\alpha)} \mathbf{L} \mathbf{D}^{-(\alpha)}$, and the transition matrix is given by

$$\mathbf{M} = [\mathbf{D}^{(\alpha)}]^{-1} \mathbf{L}^{(\alpha)}, \tag{63}$$

and $k(x_j, t|x_i) = M_{ij}^t$. Having set up the random walk on the data X, we define a cluster in the data set as a region for which the probability of escaping is low within a given time t, implying that t also plays the role of a scale parameter. The eigenmode decomposition of $\mathbf{M}^t$ yields

$$M_{ij}^t = \sum_p \lambda_p^t \psi_p(x_i) \phi_p(x_j). \tag{64}$$

Here, $\{\lambda_p\}$ is the set of eigenvalues of $\mathbf{M}$, and $\{\psi_p\}$ and $\{\phi_p\}$ are, respectively, its biorthogonal right and left eigenvectors. The spectrum of the eigenvalues decays fast and, therefore, only a few terms suffice for achieving a given relative accuracy in the sum.

Next, the diffusion distance $d_t(x_i, x_j)$, which is a measure of the closeness of data points x_i and x_j that are connected in the observation space, is defined by

$$d_t^2(x_i, x_j) = \sum_y \frac{[k(y, t|x_i) - k(y, t|x_j)]^2}{\phi_0(y)}, \tag{65}$$

where $\phi_0(y)$, the first left eigenvector of $\mathbf{M}$, is the stationary distribution of the Markov chain. $d_t(x_i, x_j)$ is computed by

$$d_t^2(x_i, x_j) = \sum_p \lambda_p^{2t} [\psi_p(x_i) - \psi_p(x_j)]^2, \tag{66}$$

so that the eigenvectors are used as a new set of coordinates for the data. The diffusion map is then defined by

$$\Psi_t(x) = \left[\lambda_1^t \psi_1(x), \lambda_2^t \psi_2(x), \cdots, \lambda_p^t \psi_p(x)\right], \tag{67}$$

where p is the number of terms that are used since, as discussed above, the spectrum of the eigenvalues decays and, therefore, one does not need to use a large number of eigenvalues and eigenvectors. Equation (67) represents the diffusion map obtained from the original data reduced to a p-dimensional space, embedded in the original space and, hence, dimensionality reduction of the system and its data has been achieved. Nadler *et al.* [172] showed that

$$d_t^2(x_i, x_j) \approx \|\Psi_t(x_i) - \Psi_t(x_j)\|^2, \tag{68}$$

Thus, the entire process may be summarized as follows. Given the similarity matrix $\mathbf{L}$, (i) normalize it using the parameter α by $\mathbf{L}^{(\alpha)} = \mathbf{D}^{-\alpha} \mathbf{L} \mathbf{D}^{-\alpha}$; (ii) normalize $\mathbf{M}$ according to, $\mathbf{M} = [\mathbf{D}^{(\alpha)}]^{-1} \mathbf{L}^{(\alpha)}$; (iii) compute the p largest eigenvalues of $\mathbf{M}^t$ and the corresponding eigenvectors, and (iv) compute the embedding $\Psi_t(x)$.

1. Example: Large-scale connectivity in default-mode networks

Diffusion maps have found many applications. For example, Margulies *et al.* [173] used the maps to analyze the organization of large-scale connectivity in the default-mode network at the opposite end of a spectrum from primary sensory and motor regions. They used connectivity data for the human and macaque monkey brains that are openly available, and utilized diffusion maps to recover a low-dimensional embedding from high-dimensional (original) connectivity data. Cortical points that are strongly connected by either many connections or few very strong connections are close in the embedding space, whereas those without connections are far apart, and diffusion maps can project such data onto a low-dimensional embedding. The diffusion maps that Margulies *et al.* [173] used corresponded to the parameter (see above) $\alpha = 1/2$, because the resulting diffusion maps (a) preserved the global relations between data points in the embedded space, (b) were more robust to noise in the connectivity matrix (see above) than other techniques, and (c) were less sensitive to the distribution of the connectivity data. In addition, (d) the resulting decreasing eigenvalues are indicative of natural ordering of the diffusion process, with the largest eigenvalues corresponding to the slowest processes and, therefore, representing the slowest variance in the connectivity patterns, and (e) using local distances, the diffusion maps address the curse of dimensionality (see above) problem, because smaller distances are more meaningful than larger ones as the number of dimensions increases.

F. Kernel analog forecasting

Kernel analog forecasting (KAF) does not discover a governing equation for a dynamic system, given a set of data, but make predictions for the future state of the system, given some of its past history. Edward Norton Lorenz was the first to introduce analog forecasting [174] for predicting dynamical systems based on historical data. In his method, one identifies an *analog*, the state in the historical data that resembles most closely the current initial data. To make prediction for the desired lead time, the historical evolution of that state is followed, and the prediction is made for the quantity of interest based on its value on the analog. Although, by construction, analog forecasting has no model error, its predictions cannot be continuous with respect to initial data and, thus, they are not physical. Burov, Giannakis and coworkers [175,176] developed and improved significantly Lorenz's original method, dubbed KAF, which we describe briefly following Burov *et al.* [176].

As usual, we consider a set of n ordered data points $X = \{x_i\} \in \mathcal{X}$, $i = 0, 1, \cdot, n-1$. Here, X is a continuous-time process, derived from Markovian dynamics for a coupled pair (X, Y) that evolve in the larger state space $\mathcal{X} \times \mathcal{Y}$, and $x_i = X(n\Delta t)$, with Δt being the sampling rate. The prediction lead time τ is assumed to be an integer multiple of Δt, $\tau = q\Delta t$. The data include the values of the associated prediction observable $F = \{f_{n+q}\}$ advanced by τ time units, which is defined by the Markovian dynamics through an unknown map $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, with $f_n = F(x_n, y_n)$.

Given initial data X and lead time τ, the KAF algorithm takes averages over values of the τ-shifted observable, provided in the training data and weighted by a kernel, $p : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, which determines how much weight should be given to a time series, initialized at point x_n, according to its proximity to x, the desired initial point, and is constructed from the data. Viewing the data-driven predictor as a map $Z_\tau : \mathcal{X} \to \mathbb{R}$, which takes initial condition x as the input, the KAF computes the map by

$$Z_\tau(x) = \frac{1}{n}\sum_{i=1}^{n-1} p(x, x_i) f_{i+q}, \tag{69}$$

with

$$p(x, x_i) = \sum_{j=0}^{\ell(\tau)-1} \frac{\psi_j(x)\phi_j(x_n)}{\sqrt{\lambda_j}}. \tag{70}$$

ϕ_j are computed by an eigenvalue problem associated with a data-driven approximation of a kernel integral operator (see below) constructed from x_n, which, in the limit of large data, provides an orthonormal basis for the entire space. $\ell(\tau)$ is a truncation parameter with $\ell(\tau) \leqslant L$, L being the desired maximum number of eigenvectors (see below). ψ_j is an out-of-sample extension of ϕ_j, computed by the Nyström method (usually called the quadrature method, which computes the numerical solution of an integral equation by replacing the integral with a representative weighted sum), orthonormalized with respect to an underlying *reproducing kernel Hilbert space structure*, a Hilbert space endowed with the inner product, $\langle f_1, f_2\rangle = \int f_1(x) f_2(x) dx$. Recall that a Hilbert space allows one to generalize the methods of linear algebra and calculus from finite-dimensional Euclidean vector spaces to infinite-dimensional ones.

Note that an important issue addressed by KAF algorithm is as follows. Because the Y component of the system is not observed, the sequences $\{x_i\}$ and $\{f_{n+q}\}$ are non-Markovian, as a result of which the idea of constructing a Markov chain from the data, which is widely used, is not the most suitable. However, the KAF algorithm evaluates a conditional expectation of the forecast conditioned, using the observed data $\{x_i\}$, explicitly incorporating information loss that results from unobserved Y.

In practice, as the first step, one must select a kernel $\kappa_n(x, x')$. How this is done is discussed in detail by Alexander and Giannakis [177]. One must also select L, the aforementioned desired maximum number of eigenvectors. Then, given κ_n, one computes a matrix $\mathbf{K}$ with $K_{ij} = \kappa_n(x_i, x_j)/n$. Next, one computes $\mathbf{v}_n = \mathbf{KI}$, and $\mathbf{w}_n = \mathbf{KV}^{-1}\mathbf{I}$, with $\mathbf{V} = \mathrm{diag}(\mathbf{v}_n)$, and determines a normalized kernel matrix, $\mathbf{S} = \mathbf{V}^{-1}\mathbf{KW}^{-1/2}$, where $\mathbf{W} = \mathrm{diag}(\mathbf{w}_n)$. The L largest singular values of $\mathbf{S}$ (i.e., the nonnegative square roots of the eigenvalues of $\mathbf{SS}^{\mathrm{T}}$), $\sigma_0, \sigma_1, \cdots, \sigma_{L-1}$, and the corresponding left singular vectors $\phi_0, \phi_1, \cdots, \phi_{L-1}$ (i.e., the eigenvectors of $\mathbf{SS}^{\mathrm{T}}$) are determined. The eigenvectors are then stacked columnwise to form a matrix $\mathbf{\Psi} = [\phi_0, \cdots, \phi_{L-1}]$. Setting $\lambda_j = \sigma_j^2$, the diagonal matrix $\mathbf{\Lambda} = \mathrm{diag}(\lambda_j)$ is formed next. The basis functions $\psi(x)$ are then computed by

$$\psi_j(x) = \frac{1}{n\sqrt{\lambda_j}}\sum_{i=0}^{n-1} \kappa(x, x_i)\phi_j(x_i). \tag{71}$$

V. TYPE-III SYSTEMS

Next, we describe emerging approaches for analyzing Type-III systems, which are those for which a large amount of data for a complex phenomenon in the systems may be available, but little is known about the governing equations for the phenomenon. Traditional approaches to analyzing such data rely on statistical methods and calculating various moments of the data, which in many cases are severely limited. There are several emerging approaches to address this problem.

A. Symbolic regression

While regression of numerical data and fitting them to an equation to better understand their implications is an old method, discovering the governing equations that describe the physics of a phenomenon for which data are available, which are typically based on ordinary and partial differential equations (ODEs and PDEs), involves manipulation of symbols and mathematical functions, such as derivatives and, therefore, represents a new type of regression. These methods, described in this and subsequent subsections, also involve stochastic optimization for deriving the governing equations.

One of the first efforts for such systems was reported in the seminal papers of Bongard and Lipson [178] and Schmidt and Lipson [179]. As the former authors stated, "A key challenge [to addressing the problem of having data but no governing equation], however, is to uncover the governing equations automatically, merely by perturbing and then observing the system in intelligent ways, just as a scientist would do in the presence of an experimental system. Obstacles to achieving this lay in the lack of efficient methods to search the space of symbolic equations and in assuming that precollected data are supplied to the modeling process." Since symbolic equations are typically in the form of ODEs and PDEs, the search space is quite large.

Bongard and Lipson [178] described a method dubbed *symbolic regression*, which consisted of three key elements: (a) *partitioning*, by which the governing equations that describe each of the system's variables are synthesized separately, even though their behaviors may be coupled, hence reducing significantly the search space. (b) *Automated probing* that, in addition to modeling, automates (numerical) experimentation, leading to an *automated scientific process*, and (c) *snipping*, which automatically simplifies and restructures models as they are synthesized to increase their accuracy, accelerate their evaluation, and make them more comprehensible for users. An automated scientific process tries [180] to mimic what many animals do, i.e., preserving the ability to operate after they are injured by creating qualitatively different compensatory behaviors

In the symbolic regression algorithm [178,179] the partitioning is carried out by a stochastic optimization approach, of which there are many [73], such as simulated annealing [181] and the genetic algorithm [182]. Such methods are efficient enough for searching a relatively large space composed of building blocks of ODEs or PDEs, if the size of the dataset is not exceedingly large. Bongard and Lipson [178] utilized the hill climbing method [183] for the optimization, a technique in which one begins the process with an arbitrary solution, and then iterates it to generate a more accurate solution by making incremental changes to the last iterate. When a differential equation for a variable i is integrated numerically by, for example, a Runge-Kutta method, references to other variables are replaced by actual data. Bongard and Lipson [178] only tried to discover a set of first-order ODEs that governed the dynamics of the system that they studied.

In symbolic regression a model consists of a set of nested expressions in which each expression i encodes the equations that describe the dynamic evolution of variable i. One also provides a set of possible mathematical operators, such as $\exp(\cdot)$, $\sin(\cdot)$, d/dx, etc., as well as operands that could be used to compose equations. During the first time step of integrating the ODEs, each operand in each equation is set to be the initial conditions and the expression is evaluated, with the output being the derivative computed for that variable. The number of times that each model is integrated is the same as the number of times that the system has been supplied with initial conditions, and all the models are optimized against all the time series observed or collected for the system.

There are at least three problems associated with symbolic regression. One is that it is computationally expensive since, in general, optimization typically requires intensive computations [73], unless certain "tricks" can be developed to accelerate them [73]. The second problem is the limitation of the approach by the number of mathematical operations and their various combinations that it can carry out. The third shortcoming of the approach is that it could be prone to overfitting, unless one carefully balances model complexity with predictive power.

B. Symbolic regression and genetic programming

Improvements to the original symbolic regression approach are emerging. In particular, a genetic programming (GP) approach, dubbed GPSR, which is a form of symbolic regression, has emerged recently that offers much promise. The GPs represent a kind of genetic algorithm in which models are represented as (nested) variable-length tree structures representing a program, instead of a fixed-length list of operators and values.

We first recall that the genetic algorithm uses concepts from genetics and the Darwinian evolution to generate possible solutions for an optimization problem, and involves four steps [73]: (a) *selection* for generating the solutions; (b) *design* of the "genome" to constrain the variables that define a possible solution, and the generation of the "phenotype" (the set of observable characteristics of a solution that result from its interaction with the environment); (c) the *crossover* and *mutation* operations that are used for generating new approximate solutions and approaching the true optimal state, and (d) *elitism*, which selects the solutions that have the potential of eventually leading to the global optimal state. A *generation* of the computations is completed after the four steps of operations have been carried out.

Similar to the theory of evolution according to which species that can adapt to their environment produce the next generation of their offsprings—the updated species in genetic algorithm—in an optimization problem solved by genetic algorithm each species, which is the set of all the parameters or,

in the present problem, the model represented by an ODE or PDE to be discovered based on reproducing the given data, are selected by evaluating the cost function or, more generally, a *fitness function*, which is a measure of the quality and/or accuracy of the solution. Each possible solution is represented by a string of numbers, or "chromosomes," and after each round of testing or simulation, one deletes a number of the worst possible solutions, and generates new ones from the best possible solutions. Therefore, a figure of merit or fitness is attributed to each possible solution that measures how close it has come to meeting the overall specification. This is done by applying the fitness function to the simulation results obtained from that possible solution. The species with a smaller cost function, or better fitness, has a higher probability of producing one or more offsprings, i.e., possibly more accurate solutions in the form of ODEs or PDEs, for the next generation, which is usually referred to as the *population*.

Using the population of the species, one solves the proposed ODE or PDE, computes the properties for which data are given, and evaluates the cost or fitness function, to choose the ODE or PDE that is more likely to produce more accurate, next generation predictions for the data. Such candidates are randomly recombined—the crossover step—and permuted—the mutation step—to generate new candidate equations. The candidates with the highest cost function, or the poorest fitness, are eliminated from the population, a step that represents natural selection in Darwinian evolution.

1. Example: Anomalous diffusion in heterogeneous media

An illuminating example is a very recent application of GPSR [184] to anomalous diffusion [171,185] in the incipient percolation cluster at the percolation threshold [72,186], which is a fractal and macroscopicaaly heterogeneous structure at all the length scales with a fractal dimension D_f whose values in 2D and 3D are, respectively, $91/48 \simeq 1.9$ and ≈ 2.53. The cluster has been used as a model of heterogeneous porous media. Diffusion in the cluster is anomalous [185]; that is, the mean-squared displacement of diffusing particles grows with time as [171,185,187], $\langle R^2(t)\rangle \propto t^{\alpha}$, where $\alpha = 2/D_w$, with D_w being the fractal dimension of the walk with, $D_w \simeq 2.87$ and 3.8 in 2D and 3D. An important, and for over two decades controversial, issue was the governing equation for $P(\mathbf{r}, t)$, the average probability that a diffusing particle is at position $\mathbf{r}$ at time t, for which various equations [188–190] were suggested.

Using random walk simulation of diffusion on the incipient percolation cluster in 2D, Im *et al.* [184] collected extensive numerical data for $P(\mathbf{r}, t)$. When they applied the GPSR method to the data, they discovered that the governing equation for $P(\mathbf{r}, t)$ is given by

$$\frac{\partial^{0.62} P}{\partial t^{0.62}} = \frac{0.82}{r}\frac{\partial P}{\partial r} + \frac{\partial^2 P}{\partial r^2}, \tag{72}$$

where $\partial^{\alpha}/\partial t^{\alpha}$ indicates fractional derivative. Note that the factor $1/r$ in the first term of the right-hand side of Eq. (72) was discovered by the algorithm, and was not included in the set of trial searches. The governing equation for $P(r, t)$, derived by Metzler *et al.* [190], is given by

$$\frac{\partial^{\alpha} P}{\partial t^{\alpha}} = \frac{1}{r^{d_s-1}}\frac{\partial}{\partial r}\left[r^{d_s-1}\frac{\partial P(r,t)}{\partial r}\right] = \frac{d_s-1}{r}\frac{\partial P}{\partial r} + \frac{\partial^2 P}{\partial r^2}, \tag{73}$$

where $d_s = 2D_f/D_w$, with $\alpha \approx 0.7$. Thus, the discovered equation and one that is generally accepted to govern anomalous diffusion in the incipient percolation cluster at the percolation threshold are practically identical.

He *et al.* [191] showed that the dynamics of transport processes in heterogeneous media that are described by a fractional diffusion equation is not self-averaging, in that time and ensemble averages of the observables, such as the mean-squared displacements, do not converge to each other. This is consistent with what is known for diffusion on the critical percolation cluster at the percolation threshold [192,193], for which the distribution of the displacements of the diffusing particle does not exhibit self-averaging. The discovery of a fractional diffusion equation for diffusion on the critical percolation cluster at the percolation threshold is fully consistent with this picture, and indicates the internal consistency accuracy of the approach.

The GPSR has also been used to discover morphology-dependent plasticity models for additively manufactured Inconel 718 [194]. Although the genetic algorithm is amenable to parallel processing and computations, at this point the GPSR is not since it involves numerically solving a population of ODEs or PDEs. Thus, one needs to develop more efficient ways of solving them to turn GPSR into a powerful and reliable tool for discovering the governing equations for complex phenomena in highly heterogeneous media.

C. Sparse identification of nonlinear dynamics

As an important improvement and extension to the original symbolic regression algorithm, Brunton *et al.* [195] proposed a method, the sparse identification of nonlinear dynamics (SINDy). Sparse regression, used for discovering the fewest terms in the governing equations that are required for accurately representing the data, avoids overfitting that often occurs in such approches. Brunton *et al.* [195] considered dynamical systems of the type expressed by Eq. (23). One then collects [195] a time history of the state $\mathbf{x}(t)$ and either measures the derivative $d\mathbf{x}(t)/dt = \dot{\mathbf{x}}(t)$ or approximates it numerically. The data are then sampled at several times t_1, $t_2, \cdots, t_m$ and organized into two matrices, given by

$$\mathbf{X} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \vdots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{bmatrix} \tag{74}$$

and

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_n(t_2) \\ \vdots & \vdots & \vdots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \cdots & \dot{x}_n(t_m) \end{bmatrix}. \tag{75}$$

One then sets up a library $\mathcal{L}(\mathbf{X})$ of candidate nonlinear functions of the columns of $\mathbf{X}$, with each column representing a

candidate function for the right-hand side of Eq. (23). There is, of course, complete freedom in selecting the candidate functions. For example,

$$\mathcal{L}(\mathbf{X}) = [1 \;\; \mathbf{X} \;\; \mathbf{X}^{(2)} \;\; \mathbf{X}^{(3)} \;\; \cdots \;\; \sin(\mathbf{X}) \;\; \cos(\mathbf{X}) \;\; \cdots], \quad (76)$$

where $\mathbf{X}^{(n)}$ denotes a polynomial of order n. Thus, for example,

$$\mathbf{X}^{(2)} = \begin{bmatrix} x_1^2(t_1) & x_1(t_1)x_2(t_1) & \cdots & x_2^2(t_1) & \cdots & x_n^2(t_1) \\ x_1^2(t_2) & x_1(t_2)x_2(t_2) & \cdots & x_2^2(t_2) & \cdots & x_n^2(t_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^2(t_m) & x_1(t_m)x_2(t_m) & \cdots & x_2^2(t_m) & \cdots & x_n^2(t_m) \end{bmatrix}. \quad (77)$$

If we know, for example, that only a few of the nonlinearities are active in each row of the $\mathbf{f}(\mathbf{x})$ in Eq. (23), then we set up a sparse regression problem to determine the sparse vectors of the coefficients, $\boldsymbol{\Xi} = [\boldsymbol{\chi}_1, \boldsymbol{\chi}_2 \cdots \boldsymbol{\chi}_n]$, which determine which nonlinearities are active:

$$\dot{\mathbf{X}} = \mathcal{L}(\mathbf{X})\boldsymbol{\Xi}. \quad (78)$$

Each column of $\boldsymbol{\chi}_k$ is a sparse vector of coefficients that determines the terms that are active on the right-hand side of Eq. (23). After $\boldsymbol{\Xi}$ is determined, a model for each row of the governing equations is constructed by

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}_k(\mathbf{x}) = \mathcal{L}(\mathbf{x}^{\mathrm{T}})\boldsymbol{\chi}_k. \quad (79)$$

In Eq. (79) $\mathcal{L}(\mathbf{x}^{\mathrm{T}})$ is a vector of symbolic functions of elements of $\mathbf{x}$, whereas $\mathcal{L}(\mathbf{X})$ is the data matrix. In other words,

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}) = \boldsymbol{\Xi}^{\mathrm{T}}[\mathcal{L}(\mathbf{x}^{\mathrm{T}})]^{\mathrm{T}}. \quad (80)$$

Note that each column of Eq. (79) requires a separate optimization to determine the sparse vector $\boldsymbol{\chi}_k$ for the kth-row equation. In general, $\mathcal{L}(\mathbf{X})$ is a $m \times p$ matrix, with p being the number of candidate functions. Naturally, $m \gg p$ because, typically, there are far more data than functions. Since the number of functional forms can be very large, one tests many different function bases and uses the sparsity and accuracy of the resulting model as the criterion for determining the correct basis to represent the data. The testing can be guided by knowledge about the physics of the problem.

It should be clear that the success of the application of the method to any phenomena and the accuracy of the resulting model depend on the choice of measurement variables, quality of the data, and the sparsifying function basis. While it may be difficult to know the correct variables *a priori*, time-delay coordinates often provide useful variables from a time series [196,197]. In this method, vectors in a new space, referred to as the embedding space, are formed from time-delayed values of the measurements,

$$s_m = [s_{n-(d-1)\tau}, s_{n-(d-2)\tau}, \cdots, s_n], \quad (81)$$

where d is the embedding dimension, and τ is the time lag or delay. According to Takens [196], if a sequence $\{s_m\}$ consists of measurements of the state of a dynamical system, then, under certain generic assumptions, the time-delay embedding provides a one-to-one image of the original set, if d is large enough. If one has m available measurements, then the number of embedding vectors is only $m - (d-1)\tau$. Of course, knowledge about the physics of the phenomena of interest also helps one to identify reasonable choices of the nonlinear functions and measurement coordinates. For example, problems in hydrodynamics have to do with the momentum conservation equation, and for Newtonian fluids with the Navier-Stokes equations.

For many important problems in science and engineering, such as those in hydrodynamics and transport and deformation in heterogeneous materials, the phenomena of interest are represented by PDEs that contain a few spatial variables, and involve either a very large number of measured data, or numerical data obtained from microscale simulations. Straightforward application of the method to such problems will be impractical, since the factorial growth of the library $\mathcal{L}$ with m in Eq. (77) and the required number of separate optimizations would make such applications impractical. But a solution has also been developed. Consider, for example, a fluid flow problem in 3D space, governed by the Navier-Stokes equations. One can use the proper orthogonal decomposition technique [198] that reduces the complexity of intensive numerical simulations that, in the present context, implies that the Navier-Stokes equations are replaced by simpler models that require much less computations to solve numerically; see also the above section on reduced dimensionality.

1. Example: Vortex shedding behind a cylinder

An illuminating application of the SINDy was made by Brunton *et al.* [195] to the classical problem of vortex shedding—oscillatory flow that occurs when a fluid flows past a bluff body at certain velocities, which depend on the body's size and shape—behind a cylinder. It was suggested a long time ago [199] that turbulent flow arises as a result of a series of Hopf bifurcations—a critical point at which, as a parameter changes, a system's stability switches and a periodic solution emerges—representing cubic nonlinearities. The cubic nonlinearity was puzzling because the Navier-Stokes equations contain only quadratic nonlinearity (the equations are second-order PDEs). When the first Hopf bifurcation was actually discovered [200,201] during the transition from a steady laminar wake to laminar periodic vortex shedding at Reynolds number, $\mathrm{Re} = 47$, it was shown [202] that a coupling between oscillatory modes and the base flow gives rise to a slow manifold that results in algebraic terms that approximate cubic nonlinearities on slow timescales.

Using data obtained by numerical simulation of the Navier-Stokes equations past a cylinder at a Reynolds number $\mathrm{Re} = 100$ reported by Colonius and Taira [202], Brunton *et al.* [195] showed that their approach recovers the Hopf normal form, a problem that had taken 30 years to resolve. Since the Navier-Stokes equations contain quadratic nonlinearity, Brunton *et al.* had to use a mean-field model with a separation of timescales, such that a fast mean-field deformation was slave to the slow vortex shedding dynamics. Thus, they used a reduced-order mean-field model for the cylinder dynamics, proposed by Noack *et al.* [163], given by Eqs. (58)–(60), with $(x_1, x_2, x_3) \to (x, y, z)$. For large values of λ in Eq. (60), the z dynamics would be slow and, therefore, the mean flow would

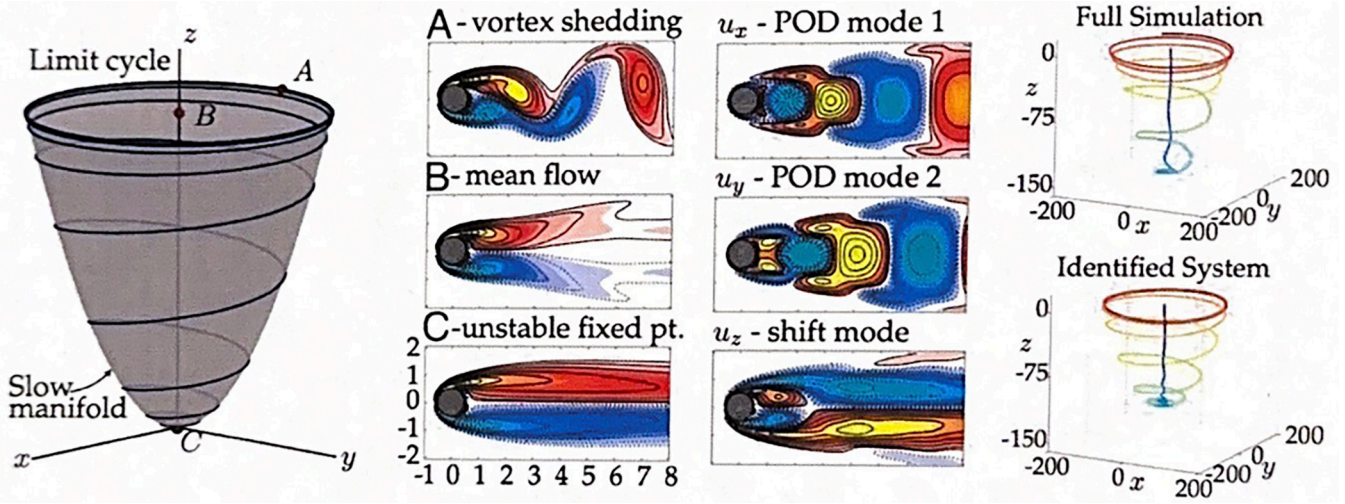

FIG. 15. The vortex shedding past a cylinder is the result of a Hopf bifurcation. Because the Navier-Stokes equations have quadratic nonlinearity, one must use a mean-field model with a separation of timescales, where a fast mean-field deformation is slave to the slow vortex shedding dynamics. The parabolic slow manifold is shown (left), with the unstable fixed point (C), mean flow (B), and vortex shedding (A). A proper orthogonal decomposition (POD) basis and shift mode were used to reduce the dimension of the problem (middle right). The identified dynamics closely match the true trajectory in the POD coordinates, and capture the quadratic nonlinearity and timescales associated with the mean-field model. Based on Ref. [195].

rapidly correct and be on the slow manifold, $z = x^2 + y^2$, given by the amplitude of vortex shedding. The Hopf normal form is recovered by substituting the algebraic forms into Eqs. (58) and (59).

Given the time history of the three coordinates, the SINDy algorithm correctly identified quadratic nonlinearities (in the Navier-Stokes equations) and reproduced a parabolic slow manifold. Equations (58)–(60) involve the derivatives whose measurements were not available, but were computed from the state variables. More importantly, when the training data do not include trajectories that originate off of the slow manifold, the algorithm *incorrectly* identifies cubic nonlinearities, hence failing to identify the slow manifold.

Figure 15 presents the results and compares them with full simulations. The parabolic slow manifold is shown on the left-hand side of Fig. 15, which contains vortex shedding indicated by A, the mean flow indicated by B, and an unstable fixed point, denoted by C. A proper orthogonal decomposition basis and shift mode were used to reduce the dimension of the problem, shown in the middle right of the figure. The agreement between the identified dynamics and the true trajectory in the proper orthogonal decomposition coordinates is excellent. The identified dynamics also captures the quadratic nonlinearity and timescales associated with the mean-field model.

The open source software package [203] PySINDy [Pyton SINDy] has been developed in Python to integrate the various versions of SINDy [204]. Note that by promoting sparsity, SINDy solves an over-determined set of equations, $\mathbf{Ax} = \mathbf{b}$, making it modular and, hence, amenable to computational innovations. Compared with the original symbolic regression described above, SINDy is extremely efficient computationally, requiring orders of magnitude less computation time. It may also be used with neural networks that provide automatic differentiation [205,206], and learning coordinates and models jointly [207,208]. Even though the approach has been applied to a wide variety of problems [117,209–230] over the past few years, and certain improvements have been made in it [231,232], it is still evolving to make it applicable to a wider class of problems, as well as making it faster computationally.

A distinct version of SINDy, weak sparse identification of nonlinear dynamics (WSINDy), first proposed by Schaeffer and McCalla [233] and improved significantly by Messenger and Bortz [234], attempts to bypass computations of the derivatives required by SINDy, hence increasing significantly the speed of the computations. The approach assumes that the function $\mathbf{f}(\mathbf{x})$ in Eq. (23) can be accurately represented by polynomials, $F(x) = x^{j-1}$, and utilizes a number of feature vectors that are large enough to include all the terms present in the underlying system. Each feature vector $v_j(x, t_k)$ is approximated by using piecewise constant quadrature,

$$v_j(x, t_k) = \int_0^{t_k} F_j[x(t)]dt \approx \Delta t \sum_{l=1}^{k} F_j[x(t_l)], \tag{82}$$

with $k = 1, 2, \cdots, K$, $v_j(x, t_0) = v_j(x, 0) = 0$, and K and Δt being, respectively, the number of discrete time steps, and the size of time steps. The quadrature yields a close approximation to the noiseless $\mathbf{x}(t)$ without smoothing, and effectively calculates a scaled expectation E for a sum of random variables of the form, $x^n\eta^p$, $E(x^n\eta^p) = E(x^n)E(\eta^p)$. Decoupling the two expected values is permitted since the noise is sampled independently of the data. Thus, many of the noise-dependent cross terms are essentially zero, if piecewise constant quadrature is used to approximate the feature vector.

By eliminating pointwise derivative approximations, one obtains estimates for the model's coefficients from noise-free data with machine precision, as well as robust identification of the PDEs with large noise in the data. One discretizes a convolutional weak form of the PDE, and utilizes separability of the test functions for efficient model identification using fast Fourier transform. Messenger and Bortz [234] showed that WSINDy algorithm has, at worse, a computational complexity

on the order of $\mathcal{O}(N^{d+1}\log N)$ for N data points in each of $d+1$ dimensions, i.e., $\mathcal{O}(\log N)$ operations per data point. The approach has been used to study a number of important problems involving complex phenomena [235–238].

Even though the applications of SINDy have so far mostly remained limited to those systems that do not have spatial heterogeneities, the approach does have applications to such systems. Turbulent flow in porous media is one example in which the interplay between the spatial heterogeneities and vortical structures gives rise to a rich variety of behavior [239]. The key question is, what is the governing equation for turbulent flow, or more generally nonlinear flows (as opposed to Darcy flow at very low Reynolds numbers that is linear), in heterogeneous porous media? This problem has yet to find a solution. Another possible application is modeling of dynamic iEEG data (see above) for brain, a system with highly complex spatial structure.

D. Machine-learning approaches

We already described the work of Lusch *et al.* [156] that used deep learning to extract the eigenfunctions of the Koopman operator. In addition, there is an emerging class of data-driven approaches for discovering the governing equations for complex phenomena that relies partly on such algorithms. A good discussion of the issues that one must address when using machine learning to discover the governing equation for a dynamical system is given by Qin *et al.* [240].

One example of such approaches is the work of DiPeitro *et al.* [241], who introduced a model for deriving the Hamiltonian of a dynamical system based on data. Suppose that the Hamiltonian is described by $\mathbf{q} = (q_1, q_2, \cdots, q_n)$ and $\mathbf{p} = (p_1, p_2, \cdots, p_n)$, where $\mathbf{q}$ and $\mathbf{p}$ represent, respectively, the position and momentum of "object" i in the system. As usual, the evolution of the system is described by, $d\mathbf{p}/dt = -\partial\mathcal{H}/\partial\mathbf{q}$ and $d\mathbf{q}/dt = \partial\mathcal{H}/\partial\mathbf{p}$, where $\mathcal{H}$ is the Hamiltonian, or total energy, of the system, subject to the initial conditions $\mathbf{q}_0$ and $\mathbf{p}_0$. The time evolution is symplectomorphic, i.e., it conserves the volume form of the phase space and the symplectic 2-form wedge product $d\mathbf{p} \wedge \mathbf{x}$. DiPietro *et al.* [241] assumed that the Hamiltonian is separable, i.e., it can be written as, $\mathcal{H} = E_p + E_k$, with E_p and E_k being the potential and kinetic energy.

Their approach, which they dubbed *sparse symplectically integrated neural network*, utilizes two neural networks, $\mathcal{N}_{E_p}$ and $\mathcal{N}_{E_k}$, which parametrize the potential and kinetic energies of the total Hamiltonian. Each network carries out a sparse regression (see above) within a search space specified by the user, which can include various functional forms, such as multivariate polynomials, trigonometric functions, and others, and computes the terms of the basis functions within the forward pass. The transformation must happen within the networks to enable the user to automatically compute gradients with respect to $\mathbf{q}$ and $\mathbf{p}$. The basis terms are then passed through a single fully connected layer, which learns its necessary terms by making the trainable parameters to be the coefficients of each basis term, which are learned linearly with respect to each term in the basis. Depending on the spacified function space, one can modify the architecture of the networks. For example, one may employ an additional layers with bias if parametrizing using trigonometric functions

For the purpose of training, as well as making predictions, the two networks are coupled with a symplectic integration scheme, which can be of any order, depending on how much computing time one is willing or can afford to spend. DiPietro *et al.* [241] used a fourth-order integration scheme. Each time the gradients of the Hamiltonian (see above) are required, it is propagated through the networks, the necessary gradients are automatically computed, and are sent to the symplectic integrator. Since, depending on the size of the time step, fourth-order symplectic integration often requires many iterations, one frequently has multiple passes through each network before the loss or cost function is computed. After the next state has been calculated, one computes the L^1-norm between the predicted and the actual next state. L^1-regularization is also incorporated so that only the essential terms of the Hamiltonian are preserved. One can also achieve the same by using thresholding that completely eliminates the nonessential terms. The loss function is then defined and computed, and the optimization process for minimizing it is carried out.

Another approach is based on deep operator networks, DeepONets [242], which learn operators accurately and efficiently from a relatively small dataset in a supervised data-driven manner. DeepONets consist of two subnetworks, one for encoding the input function at a fixed number of sensors x_i, $i = 1, \cdots, m$, which represents the branch net, and a second subnetwork for encoding the locations for the output functions, the trunk net. One performs systematic simulations for identifying the PDE that governs the data. It has been demonstrated that DeepONet significantly reduces the generalization error, when compared with the fully connected neural networks.

Note that DeepONet is different from PIML algorithms described above, which are used to make predictions for various phenomena in complex media in which the solution of a *known* PDE is modeled by a deep convolutional neural network whose parameters, together with other parameters of the model, are learned, since the fundamental underlying physics is established *a priori*. For example, Reyes *et al.* [243] used a PIML algorithm to discover viscosity models for two non-Newtonian systems, namely, polymer melts and suspensions of particles, in which they used only the data for the fluid velocity.

A hybrid method, DeepM&Mnet, a composite supervised neural network, has also been proposed that combines DeepONets with the physics encoded by PIMLs, to obtain faster and more accurate solutions for complex problems. For example, Cai *et al.* [244] developed the approach to study electroconvection that results from coupling of a flow field with an electric field, as well as the concentration distributions of the cations and anions. In their approach, given general inputs from the rest of the fields, one first pretrains DeepONets that each field predicts independently.

In another application, Mao *et al.* [245] used the same hybrid approach to study high-speed flow past a normal shock. In this phenomenon the temperature of the fluid increases rapidly, triggering chemical dissociation reactions downstream. The species give rise to appreciable changes in the properties of the fluid. Hence, one has a coupled multiphysics multiscale dynamic phenomenon. Carrying out standard numerical simulation of the phenomenon is extremely difficult,

whereas the hybrid DeepM&Mnet can integrate seamlessly, given sparse measurements of the state variables in the simulation algorithm.

VI. POSSIBLE FUTURE DIRECTIONS

The world that we live in is constantly grappling with many highly difficult but also tremendously important problems for which vast amount of data are either already available, or are becoming so, but the physical laws or, more precisely, the equations that govern them, remain elusive. They include, but not limited to, understanding the neural basis of cognition and other biological systems, predicting large earthquakes, predicting the fate of contaminants in groundwater aquifers, extracting and predicting coherent changes in the climate, understanding and predicting global soil salinization [246] as severe drought and many other factors afflict large parts of the world, managing the spread of such emerging diseases as COVID-19, controlling turbulence, and many more.

The goal of this perspective was to describe recent progress in developing theoretical and computational approaches that can meaningfully analyze complex systems in which a physical phenomenon of interest, for which some data are given, occurs. In particular, we considered those systems for which large amounts of data are available, but the governing equations for the physical phenomena of interest at the macroscale are either not known, or only partially known. As this perspective has hopefully demonstrated, many approaches have been developed. But, although the "buzzword" is that machine-learning algorithms and artificial intelligence are going to solve many, if not all the problems listed above, that is not the case, at least not for the short and intermediate timescales. Artificial intelligence is not a panacea for all problems in science and engineering, and if it is not used the right way, it can even create the misguided illusion that all the problems listed above and many more are going to be solved over the next 5–10 years, which is not the case, hence setting science back.

At the same time, as this perspective has hopefully made it clear, there has been great progress in developing approaches that not only do not rely on machine learning, but have also provided new routes for dealing with big data that are becoming available all across science and engineering. Thus, the question of which route to take is by itself a critical one to address. In some cases, such as climate modeling that involves multiple, widely disparate length scales, as well as extremely long times, the current computational power does not allow carrying out numerical simulations over all the relevant length and timescales. Therefore, a combination of machine-learning algorithms and highly resolved, but affordable simulations, is perhaps the best route. Other cases represent "either" or "or" system, whereby one can still deal with big data without resorting to machine learning, or the training a neural network with suitable architecture may be the only hope.

Even when it comes to the approaches that are currently available, while it is true that tremendous progress has been made in about a decade or so, many problems remain. Some are purely theoretical, while many are practical issues that involve the speed of the computations, the range of parameter space that can be accessed, etc. For example,

(i) although machine-learning-based approaches have enjoyed tremendus success, a rigorous theoretical foundation as to why they are successful, or when they may fail, is still lacking. Thus, one needs new theories, and perhaps new mathematics, to analyze the limitations, as well as capabilities of physics- and data-informed algorithms.

(ii) When one uses a machine-learning algorithm, the neural network is trained with some data, so that it can predict the "future" of the system if a suitable training data set is provided. But, what if one is interested in predicting the behavior of the system under circumstances for which no data are available to train the neural network? For example, suppose that the neural network is trained for predicting properties of laminar flow in a system, but one is interested in understanding and predicting how the transition from laminar flow to the turbulent regime occurs in the same system, but no data for the transition, or for the turbulent regime is used in the training, since they are not available. This is a crucial question to address, since one criticism of machine-learning approaches is that they may not be able to predict dynamics that they have never "seen." Although some progress has been made [247] based on echo state networks, much remains to be done. An echo state network uses a recurrent neural network with a sparsely connected hidden layer with typically one percent connectivity. The connectivity and weights of hidden neurons are randomly assigned, while the weights of the output neurons are learned in such a way that the network can produce or reproduce specific temporal patterns.

(iii) When it comes to the Mori-Zwanzig approach, the question of how to efficiently and accurately construct the kernel and other terms of the formulations is still very much open.

(iv) Discovering the governing equations from sparse identification of nonlinear dynamical systems still has many hurdles to overcome. One must, for example, address [176] the issue of the correct choice of measurement coordinates and of sparsifying function basis for the dynamics. There is no simple solution to this problem [195] and, therefore, a coordinated effort to incorporate expert knowledge, feature extraction, and other advanced methods is needed.

(v) Since many of the methods that were described, including symbolic regression, and machine-learning-based algorithms, involve use of stochastic optimization algorithms, one important question is whether it is possible to have no, or extremely small, training loss, when an optimization method is used. Other errors that need to be rigorously analyzed include those involved in the approximate solution of PDEs, as well as the question that is often asked, namely, does a smaller training error imply more accurate predictions?

(vi) Many multiphysics and multiscale complex phenomena occur in systems with complicated geometry that must be incorporated into the algorithms, which is not an easy task due to the required computation time. Although some efforts have been made to address such questions [248–251], much remains to be explored.

(vii) Even when it is clear that one needs a synthesis of two or more approaches, say a combination of a machine-learning algorithm and intensive numerical simulation, one needs to be equipped with, for example, optimization theory and theory of PDEs. In addition, there is always tremendous need for yet

faster numerical simulation and analysis. The combination of such branches of science is openning up new research venues.

In addition, every new approach or algorithm requires benchmarks for checking its accuracy and efficiency. When dealing with huge amounts of data for complex phenomena and systems, such benchmarks must provide a meaningful evaluation of the algorithms. Selecting such benchmarks is also not an easy task and requires careful considerations, as does the task of selecting the way by which such data should be made publicly available, a way that is accessible to a larger number of potential users.

In terms of moving in the direction of much wider use of such algorithms, we recall that one reason that platforms for conventional computations, such as OpenFOAM [252] for simulating fluid flow and transport processes, the FEniCS [253] that solves differential equations by finite-element method and the Community Atmospheric Model (CAM) [254] that is an atmospheric general circulation model, popular is that they are user-friendly. Thus, for example, in the area of applications of the physics- and data-informed algorithms or symbolic regression methods, to make such approaches "everyday tools" of research and development, they must also be user-friendly, and provide easy-to-use tools of visualization and tracking the variables as they evolve in space and time.

ACKNOWLEDGMENTS

Over the past two decades, I have benefited greatly from stimulating discussions and fruitful collaboration with many colleagues with whom I have been working on some of the problems described in this perspective. I am particularly grateful to Felipe de Barros, Jinwoo Im, Fatemeh Ghasemi, Hossein Hamzehpour, Reza Jafari, Serveh Kamrava, Joachim Peinke, Reza Rahimi Tabar, and Pejman Tahmasebi. In addition, I am grateful to the two anonymous referees, for very useful comments and suggestions that greatly improved the quality and the organization of the manuscript. I thank Niloofar Sahimi and Shayan Jalalmanesh for their invaluable help regarding preparation of the figures. The preparation of this perspective was supported in part by the National Science Foundation through Grant No. CBET 2000966, for which I am grateful.

[1] J. H. Seinfeld and S. N. Pandis, *Atmospheric Chemistry and Physics* (Wiley, New York, NY, 1998).

[2] D. Simpson, Long-period modelling of photochemical oxidants in Europe. Model calculations for July 1985, Atmos. Environ. **26**, 1609 (1992).

[3] A. Heidarinasab, B. Dabir, and M. Sahimi, Multiresolution wavelet-based simulation of transport and photochemical reactions in the atmosphere, Atmos. Environ. **38**, 6381 (2004).

[4] P. Tahmasebi, S. Kamrava, T. Bai, and M. Sahimi, Machine learning in geo- and environmental sciences: From small to large scale, Adv. Water Resour. **142**, 103619 (2020).

[5] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, Physics-informed machine learning, Nat. Rev. Phys. **3**, 422 (2021).

[6] M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Mr. Prabhat, Deep learning and process understanding for data-driven earth system science, Nature (London) **566**, 195 (2019).

[7] S. Kamrava, P. Tahmasebi, and M. Sahimi, Simulating fluid Flow in complex porous materials: Integrating the governing equations with deep-layered machines, npj Comput. Mater. **7**, 127 (2021).

[8] M. Alber, A. B. Tepole, W. R. Cannon, S. De, S. Dura-Bernal, K. Garikipati, G. Karniadakis, W. W. Lytton, P. Perdikaris, L. Petzold, and E. Kuhl, Integrating machine learning and multiscale modeling—Perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences, npj Digit. Med. **2**, 115 (2019).

[9] M. Sahimi, *Flow and Transport in Porous Media and Fractured Rock*, 2nd ed. (Wiley-VCH, Weinheim, 2011).

[10] M. Sahimi and S. E. Tajer, Self-affine distributions of the bulk density, elastic moduli, and seismic wave velocities of rock, Phys. Rev. E **71**, 046301 (2005).

[11] Z. Zhang and J. C. Moore, *Mathematical and Physical Fundamentals of Climate Change* (Elsevier, Amsterdam, 2015), Chap. 9.

[12] G. Cressman, An operational objective analysis system, Mon. Weather Rev. **87**, 367 (1959).

[13] R. E. Kalman, A new approach to linear filtering and prediction problems, Trans. ASME. J. Basic Eng. **82**, 35 (1960).

[14] G. Evensen, Using the extended Kalman filter with a multilayer quasi-geostrophic ocean model, J. Geophys. Res. **97**, 17905 (1992).

[15] G. Evensen, Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics, J. Geophys. Res. **99**, 10143 (1994).

[16] P. L. Houtekamer and H. L. Mitchell, Data assimilation using an ensemble Kalman filter technique, Mon. Weather Rev. **126**, 796 (1998).

[17] P. L. Houtekamer and H. L. Mitchell, Ensemble Kalman filtering, Q. J. R. Meteorol. Soc. **131**, 3269 (2005).

[18] H. Li, S. J. Qin, T. T. Tsotsis, and M. Sahimi, Computer simulation of gas generation and transport in landfills. VI. Dynamic updating of the model using the ensemble Kalman filter, Chem. Eng. Sci. **74**, 69 (2012).

[19] H. Li, T. T. Tsotsis, M. Sahimi, and S. J. Qin, Ensembles-based and GA-based optimization for landfill gas production, AIChE J. **60**, 2063 (2014).

[20] F. Hourdin, T. Mauritsen, A. Gettelman, J.-C. Golaz, V. Balaji, Q. Duan, D. Folini, D. Ji, D. Klocke, Y. Qian, F. Rauser, C. Rio, L. Tomassini, M. Watanabe, and D. Williamson, The art and science of climate model tuning, Bull. Am. Meteorol. Soc. **98**, 589 (2017).

[21] S. Scher, Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning, Geophys. Res. Lett. **45**, 12 (2018).

[22] P. D. Dueben and P. Bauer, Challenges and design choices for global weather and climate models based on machine learning, Geosci. Model Dev. **11**, 3999 (2018).

[23] D. C. Park, A time series data prediction scheme using bilinear recurrent neural network, in *Proceedings of the International*

Conference on Information Science and Applications (Seoul, South Korea, 2010), p. 1.

[24] R. Fablet, S. Ouala, and C. Herzet, Bilinear residual neural network for the identification and forecasting of geophysical dynamics, in *Proceedings of the 26th European Signal Processing Conference (EUSIPCO)* (Rome, Italy, 2018), p. 1477.

[25] K. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach, Phys. Rev. Lett. **120**, 024102 (2018).

[26] P. Laloyaux, M. Bonavita, M. Dahoui, J. Farnan, S. Healy, E. Hólm, and S. Lang, Towards an unbiased stratospheric analysis, Q. J. R. Meteorol. Soc. **146**, 2392 (2020).

[27] S. Rasp, M. S. Pritchard, and P. Gentine, Deep learning to represent subgrid processes in climate models, Proc. Natl. Acad. Sci. USA **115**, 9684 (2018).

[28] M. T. Giraudo and L. Sacerdote, Jump-diffusion processes as models for neuronal activity, Biosystems **40**, 75 (1997).

[29] J. Brajard, A. Carrassi, M. Bocquet, and L. Bertino, Combining data assimilation and machine learning to infer unresolved scale parametrisation, Philos. Trans. R. Soc. London **379**, 20200086 (2021).

[30] M. R. Rahimi Tabar, *Analysis and Data-based Reconstruction of Complex Nonlinear Dynamical Systems: Using the Methods of Stochastic Processes* (Springer, Bern, 2019).

[31] G. A. Pavliotis, A. M. Stuart, and U. Vaes, Derivative-free Bayesian inversion using multiscale dynamics, SIAM J. Appl. Dyn. Syst. **21**, 284 (2022).

[32] See, for example, C. M. Bishop, Neural networks and their applications, Rev. Sci. Instrum. **65**, 1803 (1994).

[33] S. Torquato, *Random Heterogeneous Materials* (Springer, New York, NY, 2002).

[34] M. Sahimi, *Heterogeneous Materials*, Vols. I and II (Springer, New York, NY, 2003).

[35] S. Kamrava, P. Tahmasebi, and M. Sahimi, Linking morphology of porous media to their macroscopic permeability by deep learning, Transp. Porous Media **131**, 427 (2020).

[36] S. Kamrava, P. Tahmasebi, and M. Sahimi, Enhancing images of shale formations by a hybrid stochastic and deep learning algorithm, Neural Netw. **118**, 310 (2019).

[37] S. Kamrava, P. Tahmasebi, and M. Sahimi, Physics- and image-based prediction of fluid flow and transport in complex porous membranes and materials by deep learning, J. Membr. Sci. **622**, 119050 (2021).

[38] S. Ioffe and C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, Proc. Mach. Learn. Res. **37**, 448 (2015).

[39] H. Hamzehpour and M. Sahimi, Development of optimal models of porous media by combining static and dynamic data: The porosity distribution, Phys. Rev. E **74**, 026308 (2006).

[40] H. Hamzehpour, M. R. Rasaei, and M. Sahimi, Development of optimal models of porous media by combining static and dynamic data: The permeability and porosity distributions, Phys. Rev. E **75**, 056311 (2007).

[41] S. Kullback and R. A. Leibler, On information and sufficiency, Ann. Math. Stat. **22**, 79 (1951).

[42] H. Andrä, N. Combaret, J. Dvorkin, E. Glatt, J. Han, M. Kabel, Y. Keehm, F. Krzikalla, M. Lee, C. Madonna, M. Marsh, T. Mukerji, E. H. Saenger, R. Sain, N. Saxena, S. Ricker, A. Wiegmann, and X. Zhan, Digital rock physics benchmarks-Part I: Imaging and segmentation, Comput. Geosci. **50**, 25 (2013).

[43] G. Kissas, Y. Yang, E. Hwuang, W. R. Witschey, J. A. Detre, and P. Perdikaris, Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physicsinformed neural networks, Comput. Methods Appl. Mech. Eng. **358**, 112623 (2020).

[44] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), Chia Laguna Resort, Sardinia, Italy*, edited by Y. W. Teh and M. Titterington (PMLR, 2010), Vol. 9, pp. 249–256.

[45] Y. Zhu, N. Zabaras, P. S. Koutsourelakis, and P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, J. Comput. Phys. **394**, 56 (2019).

[46] N. Geneva and N. Zabaras, Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks, J. Comput. Phys. **403**, 109056 (2020).

[47] J. L. Wu, K. Kashinath, A. Albert, D. Chirila, Prabhat, and H. Xiao, Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems, J. Comput. Phys. **406**, 109209 (2020).

[48] N. Sukumar and A. Srivastava, Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks, Comput. Methods Appl. Mech. Eng. **389**, 114333 (2022).

[49] A. Kashefi, D. Rempe, and L. J. Guibas, A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries, Phys. Fluids **33**, 027104 (2021).

[50] Y. LeCun and Y. Bengio, Convolutional networks for images, speech, and time series, in *The Handbook of Brain Theory and Neural Networks*, edited by M. A. Arbib (MIT Press, Cambridge, MA, 1995).

[51] J. Winkens, J. Linmans, B. S. Veeling, T. S. Cohen, and M. Welling, Improved semantic segmentation for histopathology using rotation equivariant convolutional networks, in *Proceedings of the 1st Conference on Medical Imaging with Deep Learning (MIDL)* (Amsterdam, The Netherlands 2018), https://openreview.net/pdf?id=SyXbz1hiM.

[52] T. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling, Gauge equivariant convolutional networks and the icosahedral CNN, Proc. Mach. Learning Res. **97**, 1321 (2019).

[53] H. Owhadi and G. R. Yoo, Kernel flows: from learning kernels from data into the abyss, J. Comput. Phys. **389**, 22 (2019).

[54] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Numerical Gaussian processes for time-dependent and nonlinear partial differential equations, SIAM J. Sci. Comput. **40**, A172 (2018).

[55] B. Hamzi and H. Owhadi, Learning dynamical systems from data: A simple cross-validation perspective, Part I: Parametric kernel flows, Physica D **421**, 132817 (2021).

[56] S. Wang, X. Yu, and P. Perdikaris, When and why PINNs fail to train: A neural tangent kernel perspective, J. Comput. Phys. **449**, 110768 (2022).

[57] S. Wang, H. Wang, and P. Perdikaris, On the eigenvector bias of Fourier feature networks: from regression to solving multi-scale PDEs with physics-informed neural networks, Comput. Methods Appl. Mech. Eng. **384**, 113938 (2021).

[58] H. Owhadi, Do ideas have shape? Idea registration as the continuous limit of artificial neural networks, Physica D **444**, 133592 (2023).
[59] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, Graph neural networks: A review of methods and applications, AI Open **1**, 57 (2020).
[60] A. Mathews, M. Francisquez, J. Hughes, and D. Hatch, Uncovering edge plasma dynamics via deep learning from partial observations, Phys. Rev. E **104**, 025205 (2021).
[61] D. Pfau, J. S. Spencer, A. G. Matthews, and W. M. C. Foulkes, Ab initio solution of the many-electron Schrödinger equation with deep neural networks, Phys. Rev. Res. **2**, 033429 (2020).
[62] G. P. Pun, R. Batra, R. Ramprasad, and Y. Mishin, Physically informed artificial neural networks for atomistic modeling of materials, Nat. Commun. **10**, 2339 (2019).
[63] D. Li, K. Xu, J. M. Harris, and E. Darve, Coupled time-lapse full-waveform inversion for subsurface flow problems using intrusive automatic differentiation, Water Resour. Res. **56**, e2019WR027032 (2020).
[64] S. Kamrava, J. Im, F. P. J. de Barros, and M. Sahimi, Estimating dispersion coefficient in flow through heterogeneous porous media by a deep convolutional neural network, Geophys. Res. Lett. **48**, e2021GL094443 (2021).
[65] H. Wu, W. Z. Fang, Q. Kang, W. Q. Tao, and R. Qiao, Predicting effective diffusivity of porous media from images by deep learning, Sci. Rep. **9**, 20387 (2019).
[66] L. Zhou, L. Shi, and Y. Zha, Seeing macro-dispersivity from hydraulic conductivity field with convolutional neural network, Adv. Water Resour. **138**, 103545 (2020).
[67] W. Zhu, K. Xu, E. Darve, and G. C. Beroza, A general approach to seismic inversion with automatic differentiation, Comput. Geosci. **151**, 104751 (2021).
[68] J. Behler and M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, Phys. Rev. Lett. **98**, 146401 (2007).
[69] L. Zhang, J. Han, H. Wang, R. Car, and E. Weinan, Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics, Phys. Rev. Lett. **120**, 143001 (2018).
[70] W. Jia, H. Wang, M. Chen, D. Lu, L. Lin, R. Car, E. Weinan, and L. Zhang, Pushing the limit of molecular dynamics with ab initio accuracy to 100 million atoms with machine learning, in *Proceedings of 2020 International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2020, Atlanta, Georgia* (IEEE, Piscataway, NJ, 2020).
[71] A. Nakata, J. S. Baker, S. Y. Mujahed, J. T. L. Poulton, S. Arapan, J. Lin, Z. Raza, S. Yadav, L. Truflandier, T. Miyazaki, and D. R. Bowler, Large scale and linear scaling DFT with the CONQUEST code, J. Chem. Phys. **152**, 164112 (2020).
[72] M. Sahimi, *Applications of Percolation Theory*, 2nd ed. (Springer, New York, NY, 2023).
[73] M. Sahimi and P. Tahmasebi, Reconstruction, optimization, and design of heterogeneous materials and media: Basic principles, computational algorithms, and applications, Phys. Rep. **939**, 1 (2021).
[74] P. Tahmasebi and M. Sahimi, Reconstruction of three-dimensional porous media using a single thin section, Phys. Rev. E **85**, 066709 (2012).
[75] P. Tahmasebi and M. Sahimi, Cross-correlation function for accurate reconstruction of heterogeneous media, Phys. Rev. Lett. **110**, 078002 (2013).
[76] P. Tahmasebi and M. Sahimi, Enhancing multiple-point geostatistical modeling. 1: Graph theory and pattern adjustment, Water Resour. Res. **52**, 2074 (2016).
[77] P. Tahmasebi and M. Sahimi, Enhancing multiple-point geostatistical modeling. 2: Iterative simulation and multiple distance functions, Water Resour. Res. **52**, 2099 (2016).
[78] N. Alqahtani, F. Alzubaidi, R. T. Armstrong, P. Swietojanski, and P. Mostaghimi, Machine learning for predicting properties of porous media from 2D x-ray images, J. Pet. Sci. Eng. **184**, 106514 (2020).
[79] K. M. Graczyk and M. Matyka, Predicting porosity, permeability, and tortuosity of porous media from images by deep learning, Sci. Rep. **10**, 21488 (2020).
[80] R. Mantegna and H. E. Stanley, *An Introduction to Econophysics: Correlations and Complexities in Finance* (Cambridge University Press, New York, NY, 2000).
[81] P. Manshour, S. Saberi, M. Sahimi, J. Peinke, A. F. Pacheco, and M. R. Rahimi Tabar, Turbulencelike behavior of seismic time series, Phys. Rev. Lett. **102**, 014101 (2009).
[82] P. Ch. Ivanov, L. A. Amaral, A. L. Goldberger, S. Havlin, M. G. Rosenblum, Z. R. Struzik, and H. E. Stanley, Multifractality in human heartbeat dynamics, Nature (London) **399**, 461 (1999).
[83] Y. Ashkenazy, P. Ch. Ivanov, S. Havlin, C.-K. Peng, A. L. Goldberger, and H. E. Stanley, Magnitude and sign correlations in heartbeat fluctuations, Phys. Rev. Lett. **86**, 1900 (2001).
[84] J. D. Hamilton, *Time Series Analysis* (Princeton University Press, Princeton, NJ, 1994).
[85] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis* (Cambridge University Press, London, UK, 2003).
[86] R. H. Stoffer and S. David, *Time Series Analysis and Its Applications* (Springer-Verlag, Berlin, 2006).
[87] R. Friedrich and J. Peinke, Description of a turbulent cascade by a Fokker-Planck equation, Phys. Rev. Lett. **78**, 863 (1997).
[88] J. Davoudi and M. R. Rahimi Tabar, Theoretical model for the Kramers-Moyal description of turbulence cascades, Phys. Rev. Lett. **82**, 1680 (1999).
[89] R. Friedrich, J. Peinke, M. Sahimi, and M. R. Rahimi Tabar, Approaching complexity by stochastic methods: From biological systems to turbulence, Phys. Rep. **506**, 87 (2011).
[90] B. B. Mandelbrot and J. W. van Ness, Fractional Brownian motions, fractional noises, and applications, SIAM Rev. **10**, 422 (1968).
[91] G. R. Jafari, S. M. Fazeli, F. Ghasemi, S. M. Vaez Allaei, M. R. Rahimi Tabar, A. Iraji zad, and G. Kavei, Stochastic analysis and regeneration of rough surfaces, Phys. Rev. Lett. **91**, 226101 (2003).
[92] F. Ghasemi, M. Sahimi, J. Peinke, and M. R. Rahimi Tabar, Analysis of non-stationary data for heart-rate fluctuations in terms of drift and diffusion coefficients, J. Biol. Phys. **32**, 117 (2006).
[93] H. Risken, *The Fokker-Planck Equation*, 2nd ed. (Springer, Berlin, 1996).
[94] F. Ghasemi, M. Sahimi, J. Peinke, R. Friedrich, G. R. Jafari, and M. R. Rahimi Tabar, Markov analysis and Kramers-Moyal expansion of nonstationary stochastic processes with an

application to the fluctuations in the oil price, Phys. Rev. E **75**, 060102(R) (2007).
[95] J. P. Bouchaud and R. Cont, A Langevin approach to stock market fluctuations and crashes, Eur. Phys. J. B **6**, 543 (1998).
[96] F. Ghasemi, J. Peinke, M. Sahimi, and M. R. Rahimi Tabar, Regeneration of stochastic processes: An inverse method, Eur. Phys. J. B **47**, 411 (2005).
[97] R. F. Pawula, Approximation of the linear Boltzmann equation by the Fokker-Planck equation, Phys. Rev. **162**, 186 (1967).
[98] M. Anvari, K. Lehnertz, M. R. Rahimi Tabar, and J. Peinke, Disentangling the stochastic behavior of complex time series, Sci. Rep. **6**, 35435 (2016).
[99] R. Sirovich, L. Sacerdote, and A. E. P. Villa, Cooperative behavior in a jump diffusion model for a simple network of spiking neurons, Math. Biosci. Eng. **11**, 385 (2014).
[100] E. Daly and A. Porporato, Probabilistic dynamics of some jump-diffusion systems, Phys. Rev. E **73**, 026108 (2006).
[101] R. Cont and P. Tankov, *Financial Modelling with Jump Processes* (Chapman & Hall, Boca Raton, FL, 2004).
[102] J. Prusseit and K. Lehnertz, Stochastic qualifiers of epileptic brain dynamics, Phys. Rev. Lett. **98**, 138103 (2007).
[103] K. Lehnertz, Epilepsy and nonlinear dynamics, J. Biol. Phys. **34**, 253 (2008).
[104] L. R. Gorjão, J. Heysel, K. Lehnertz, and M. R. Rahimi Tabar, Analysis and data-driven reconstruction of bivariate jump-diffusion processes, Phys. Rev. E **100**, 062127 (2019).
[105] F. Nikakhtar, L. Parkavosi, M. R. Rahimi Tabar, M. Sahimi, K. Lehnertz, and U. Feudel, Data-driven reconstruction of stochastic dynamical equations based on statistical moments, New J. Phys. **25**, 083025 (2023).
[106] M. R. Rahimi Tabar, F. Nikakhtar, L. Parkavousi, A. Akhshi, U. Feudel, and K. Lehnertz, Revealing higher-order interactions in high-dimensional complex systems: A data-driven approach, Phys. Rev. X **14**, 011050 (2024).
[107] H. Mori, Transport, collective motion, and Brownian motion, Prog. Theor. Phys. **33**, 423 (1965).
[108] R. Zwanzig, Nonlinear generalized Langevin equations, J. Stat. Phys. **9**, 215 (1973).
[109] G. F. Mazenko, *Nonequilibrium Statistical Mechanics* (Wiley-VCH, Weinheim, 2006).
[110] D. J. Evans and G. Morriss, *Statistical Mechanics of Nonequilibrium Liquids* (Cambridge University Press, Cambridge, UK, 2008).
[111] C. Hijón, P. Espãnol, E. Vanden-Eijnden, and R. Delgado-Buscalioni, Mori-Zwanzig formalism as a practical computational tool, Faraday Discuss. **144**, 301 (2010).
[112] A. J. Chorin, O. H. Hald, and R. Kupferman, Optimal prediction and the Mori-Zwanzig representation of irreversible processes, Proc. Natl. Acad. Sci. USA **97**, 2968 (2000).
[113] S. K. J. Falkena, C. Quinn, J. Sieber, J. Frank, and H. A. Dijkstra, Derivation of delay equation climate models using the Mori-Zwanzig formalism, Proc. R. Soc. A **475**, 20190075 (2019).
[114] B. O. Koopman, Hamiltonian systems and transformation in Hilbert space, Proc. Natl. Acad. Sci. USA **17**, 315 (1931).
[115] A. Gouasmi, E. J. Parish, and K. Duraisamy, *A priori* estimation of memory effects in reduced-order models of nonlinear systems using the Mori-Zwanzig formalism, Proc. R. Soc. London A **473**, 20170385 (2017).
[116] Y. Tian, Y. T. Lin, M. Anghel, and D. Livescu, Data-driven learning of Mori-Zwanzig operators for isotropic turbulence, Phys. Fluids **33**, 125118 (2021).
[117] Y. Guan, S. L. Brunton, and I. Novosselov, Sparse nonlinear models of chaotic electroconvection, R. Soc. Open Sci. **8**, 202367 (2021).
[118] W. Chu and X. Li, The Mori-Zwanzig formalism for the derivation of a fluctuating heat conduction model from molecular dynamics, Commun. Math. Sci. **17**, 539 (2019).
[119] Y. T. Lin, Y. Tian, D. Livescu, and M. Anghel, Data-driven learning for the Mori-Zwanzig formalism: A generalization of the Koopman learning framework, SIAM J. Appl. Dyn. Syst. **20**, 2558 (2021).
[120] E. J. Parish and K. Duraisamy, Non-Markovian closure models for large eddy simulations using the Mori-Zwanzig formalism, Phys. Rev. Fluids **2**, 014604 (2017).
[121] E. J. Parish and K. Duraisamy, A dynamic subgrid scale model for large eddy simulations based on the Mori-Zwanzig formalism, J. Comput. Phys. **349**, 154 (2017).
[122] S. Maeyama and T.-H. Watanabe, Extracting and modeling the effects of small-scale fluctuations on large-scale fluctuations by Mori-Zwanzig projection operator method, J. Phys. Soc. Jpn. **89**, 024401 (2020).
[123] J. Li and P. Stinis, Mori-Zwanzig reduced models for uncertainty quantification, J. Comput. Dyn. **6**, 39 (2019).
[124] P. Stinis, Higher order Mori-Zwanzig models for the Euler equations, Multiscale Model. Simul. **6**, 741 (2007).
[125] R. E. Bellman, *Dynamic Programming* (Princeton University Press, Princeton, NJ, 1957), p. ix.
[126] S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé, Data-driven model reduction and transfer operator approximation, J. Nonlin. Sci. **28**, 985 (2018).
[127] G. Snyder and Z. Song, Koopman operator theory for nonlinear dynamic modeling using dynamic mode decomposition, arXiv:2110.08442v1.
[128] P. Bevanda, S. Sosnowski, and S. Hirche, Koopman operator dynamical models: Learning, analysis and control, Annu. Rev. Control **52**, 197 (2021).
[129] G. A. Pavliotis, Stochastic Processes and Applications: Diffusion Processes, in *The Fokker-Planck and Langevin Equations* (Springer, Berlin, 2014).
[130] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition, J. Nonlin. Sci. **25**, 1307 (2015).
[131] L. Molgedey and H. G. Schuster, Separation of a mixture of independent signals using time delayed correlations, Phys. Rev. Lett. **72**, 3634 (1994).
[132] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis* (Wiley, New York, NY, 2001).
[133] G. Pérez-Hernńdez, F. Paul, T. Giorgino, G. De Fabritiis, and F. Noé, Identification of slow molecular order parameters for Markov model construction, J. Chem. Phys. **139**, 015102 (2013).
[134] C. R. Schwantes and V. S. Pande, Improvements in Markov State Model construction reveal many non-native interactions in the folding of NTL9, J. Chem. Theory Comput. **9**, 2000 (2013).
[135] A. Ziehe and K.-R. Müller, TDSEP—An efficient algorithm for blind separation using time structure, in *Proceedings*

of International Conference on Artificial Neural Networks (ICANN), edited by L. Niklasson, M. Boéden, and T. Zimeke (Springer, Berlin, 1998), p. 675.

[136] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. **656**, 5 (2010).

[137] K. K. Chen, J. H. Tu, and C. W. Rowley, Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses, J. Nonlinear Sci. **22**, 887 (2012).

[138] S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz, Compressed sensing and dynamic mode decomposition, J. Comput. Dyn. **2**, 165 (2015).

[139] S. Klus, P. Gelß, S. Peitz, and C. Schütte, Tensor-based dynamic mode decomposition, Nonlinearity **31**, 3359 (2018).

[140] M. R. Jovanović, P. J. Schmid, and J. W. Nichols, Sparsity-promoting dynamic mode decomposition, Phys. Fluids **26**, 024103 (2014).

[141] I. Mezić, Analysis of fluid flows via spectral properties of the Koopman operator, Annu. Rev. Fluid Mech. **45**, 357 (2013).

[142] F. Noé and F. Nüske, A variational approach to modeling slow processes in stochastic dynamical systems, Multiscale Model. Simul. **11**, 635 (2013).

[143] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé, Variational approach to molecular kinetics, J. Chem. Theory Comput. **10**, 1739 (2014).

[144] F. Nüske, R. Schneider, F. Vitalini, and F. Noé, Variational tensor approach for approximating the rare-event kinetics of macromolecular systems, J. Chem. Phys. **144**, 054105 (2016).

[145] A matrix $\mathbf{M}^+$ is said to be the Moore-Penrose pseudo-inverse of $\mathbf{M}$, if it satisfies three conditions: (i) $\mathbf{M}\mathbf{M}^+\mathbf{M} = \mathbf{M}$, (ii) $M^+\mathbf{M}\mathbf{M}^+ = \mathbf{M}^+$, and (iii) both $\mathbf{M}\mathbf{M}^+$ and $\mathbf{M}^+\mathbf{M}$ are Hermitian. See, for example, R. Penrose, A generalized inverse for matrices, Proc. Cambridge Philos. Soc. **51**, 406 (1955).

[146] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, A kernel-based method for data-driven Koopman spectral analysis, J. Comput. Dyn. **2**, 247 (2015).

[147] K. Mori, T. Arzberger, F. A. Grasser, I. Gijselinck, S. May, K. Rentzsch, S. M. Weng, M. H. Schludi, J. van der Zee, M. Cruts, C. Van Broeckhoven, E. Kremmer, H. A. Kretzschmar, C. Haass, and D. Edbauer, Bidirectional transcripts of the expanded C9orf72 hexanucleotide repeat are translated into aggregating dipeptide repeat proteins, Acta Neuropathol. **126**, 881 (2013).

[148] P. E. A. Ash, K. F. Bieniek, T. F. Gendron, T. Caulfield, W. L. Lin, M. Dejesus-Hernandez, M. M. van Blitterswijk, K. Jansen-West, R. Paul, R. Rademakers, K. B. Boylan, D. W. Dickson, and L. Petrucelli, Unconventional translation of C9ORF72 GGGGCC expansion generates insoluble polypeptides specific to c9FTD/ALS, Neuron **77**, 639 (2013).

[149] S. Zheng, A. Sahimi, K. S. Shing, and M. Sahimi, Molecular dynamics study of structure, folding and aggregation of poly-glycine-alanine (Poly-GA), J. Chem. Phys. **150**, 144307 (2019).

[150] S. Zheng, A. Sahimi, K. S. Shing, and M. Sahimi, Molecular dynamics study of structure, folding and aggregation of poly-proline-arginine (Poly-PR) and poly-glycine-arginine (PolGR) proteins, Biophys. J. **120**, 64 (2021).

[151] N. Djurdjevac, M. Sarich, and C. Schütte, Estimating the eigenvalue error of Markov state models, Multiscale Model. Simul. **10**, 61 (2012).

[152] F. Noé and C. Clementi, Kinetic distance and kinetic maps from molecular dynamics simulations, J. Chem. Theory Comput. **11**, 5002 (2015).

[153] The k-means algorithm tries to iteratively partition a dataset into k predefined distinct non-overlapping clusters (subgroups), with each data point belonging to only one cluster. The algorithms attempts to make the intra-cluster data points as similar as possible, while also keeping the clusters as different as possible. To do so, it assigns data points to a cluster such that the sum of the squared distances between the data points and the cluster's centroid (arithmetic mean of all the data points in that cluster) is minimum.

[154] B. E. Husic and V. S. Pande, Markov state models: From an art to a science, J. Am. Chem. Soc. **140**, 2386 (2018).

[155] A Markov state model (MSM) is set up with a series of states, and is parameterized with the rates between the states. The MSM can have a large number of states, because the states make it possible to construct a very high resolution model of the intrinsic dynamics of the system, and parameterize the model using relatively short molecular dynamics trajectories, since the kinetic distance between adjacent states is small and, thus, short simulations suffice for observing transitions between them. When developing a MSM, the main questions to be addressed are, how does one define the states in a kinetically meaningful scheme, and how can one use this state decomposition to build a transition matrix in an efficient manner; see, for example, Ref. [154].

[156] P. Deuflhard and M. Weber, Robust perron cluster analysis in conformation dynamics, Linear Algebra Appl. **398**, 161 (2005).

[157] S. Röblitz and M. Weber, Fuzzy spectral clustering by PCCA+: Application to Markov state models and data classification, Adv. Data Anal. Classif. **7**, 147 (2013).

[158] B. Lusch, J. N. Kutz, and S. L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics, Nat. Commun. **9**, 4950 (2018).

[159] C. Wehmeyer and F. Noé, Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics, J. Chem. Phys. **148**, 241703 (2018).

[160] A. Mardt, L. Pasquali, H. Wu, and F. Noé, VAMPnets: Deep learning of molecular kinetics, Nat. Commun. **9**, 5 (2018).

[161] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Syst. **2**, 303 (1989).

[162] K. Hornik, M. Stinchcombe, and H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, Neural Netw. **3**, 551 (1990).

[163] B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. A. Thiele, Hierarchy of low-dimensional models for the transient and post-transient cylinder wake, J. Fluid Mech. **497**, 335 (2003).

[164] E. Kaiser, J. N. Kutz, and S. L. Brunton, Data-driven discovery of Koopman eigenfunctions for control, Mach. Learn.: Sci. Technol. **2**, 035023 (2021).

[165] D. Giannakis, Data-driven spectral decomposition and forecasting of ergodic dynamical systems, Appl. Comput. Harmon. Anal. **47**, 338 (2019).

[166] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, Geometric diffusions as a tool for

harmonic analysis and structure definition of data: Diffusion maps, Proc. Natl. Acad. Sci. USA **102**, 7426 (2005).
[167] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, Geometric diffusions as a tool for harmonic analysis and structure definition of data: Multiscale methods, Proc. Natl. Acad. Sci. USA **102**, 7432 (2005).
[168] D. Giannakis and A. J. Majda, Nonlinear Laplacian spectral analysis for time series with intermittency and low-frequency variability, Proc. Natl. Acad. Sci. **109**, 2222 (2012).
[169] G. Froyland and K. Padberg, Almost-invariant sets and invariant manifolds—Connecting probabilistic and geometric descriptions of coherent structures in flows, Physica D **238**, 1507 (2009).
[170] Z. Trstanova, B. Leimkuhler, and T. Lelièvre, Local and global perspectives on diffusion maps in the analysis of molecular systems, Proc. R. Soc. A **476**, 20190036 (2020).
[171] B. D. Hughes, *Random Walks and Random Environments* (Oxford University Press, London, UK, 1995).
[172] B. Nadler, S. Lafon, R. R. Coifman, and I. G. Kevrekidis, Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators, in *Proceedings of the 18th International Conference on Neural Information Processing Systems*, edited by Y. Weiss, B. Schölkopf, and J. C. Platt (MIT Press, Cambridge, UK, 2005), p. 955.
[173] D. S. Margulies, S. S. Ghosh, A. Goulas, M. Falkiewicz, J. M. Huntenburg, G. Langs, G. Bezgin, S. B. Eickhoff, F. X. Castellanos, M. Petrides, E. Jefferies, and J. Smallwoodn, Situating the default-mode network along a principal gradient of macroscale cortical organization, Proc. Natl. Acad. Sci. USA **113**, 12574 (2016).
[174] E. N. Lorenz, Atmospheric predictability as revealed by naturally occurring analogues, J. Atmos. Sci. **26**, 636 (1969).
[175] Z. Zhao and D. Giannakis, Analog forecasting with dynamics-adapted kernels, Nonlinearity **29**, 2888 (2016).
[176] D. Burov, D. Giannakis, K. Manohar, and A. Stewart, Kernel analog forecasting: Multiscale test problems, Multiscale Model. Simul. **19**, 1011 (2021).
[177] R. Alexander and D. Giannakis, Operator-theoretic framework for forecasting nonlinear time series with kernel analog techniques, Physica D **409**, 132520 (2020).
[178] J. Bongard and H. Lipson, Automated reverse engineering of nonlinear dynamical systems, Proc. Natl. Acad. Sci. USA **104**, 9943 (2007).
[179] M. Schmidt and H. Lipson, Distilling free-form natural laws from experimental data, Science **324**, 81 (2009).
[180] J. Bongard, V. Zykov, and H. Lipson, Resilient machines through continuous self-modeling, Science **314**, 1118 (2006).
[181] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, Optimization by simulated annealing, Science **220**, 671 (1983).
[182] S. Katoch, S. Singh Chauhan, and V. Kumar, A review on genetic algorithm: Past, present, and future, Multimed. Tools. Appl. **80**, 8091 (2021).
[183] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. (Prentice Hall, Upper Saddle River, NJ, 2003), p. 111.
[184] J. Im, F. P. J. de Barros, S. Masri, M. Sahimi, and R. M. Ziff, Data-driven discovery of the governing equations for transport in heterogeneous media by stochastic optimization, Phys. Rev. E **107**, L013301 (2023).
[185] Y. Gefen, A. Aharony, and S. Alexander, Anomalous diffusion on percolation clusters, Phys. Rev. Lett. **50**, 77 (1983).
[186] D. Stauffer and A. Aharony, *Introduction to Perclation Theory*, 2nd. ed. (Taylor & Francis, London, UK, 1994).
[187] M. Sahimi, B. D. Hughes, L. E. Scriven, and H. T. Davis, Stochastic transport in disordered systems, J. Chem. Phys. **78**, 6849 (1983).
[188] B. O'Shaughnessy and I. Procaccia, Analytical solutions for diffusion on fractal objects, Phys. Rev. Lett. **54**, 455 (1985).
[189] M. Giona and H. E. Roman, Fractional diffusion equation for transport phenomena in random media, Physica A **185**, 87 (1992).
[190] R. Metzler, W. G. Glöckle, and T. F. Nonnenmacher, Fractional model equation for anomalous diffusion, Physica A **211**, 13 (1994).
[191] Y. He, S. Burov, R. Metzler, and E. Barkai, Random time-scale invariant diffusion and transport coefficients, Phys. Rev. Lett. **101**, 058101 (2008).
[192] A. Pacheco-Pozo and I. M. Sokolov, Universal fluctuations and ergodicity of generalized diffusivity on critical percolation clusters, J. Phys. A: Math. Theor. **55**, 345001 (2022).
[193] A. Bunde and J. Dräger, Localization in disordered structures: Breakdown of the self-averaging hypothesis, Phys. Rev. E **52**, 53 (1995).
[194] K. Garbrecht, M. Aguilo, A. Sanderson, A. Rollett, R. M. Kirby, and J. Hochhalter, Interpretable machine learning for texture-dependent constitutive models with automatic code generation for topological optimization, Integr. Mater. Manuf. Innov. **10**, 373 (2021).
[195] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. USA **113**, 3932 (2016).
[196] F. Takens, Detecting strange attractors in turbulence, Lect. Notes. Math. **898**, 366 (1981).
[197] H. Ye, R. J. Beamish, S. M. Glaser, S. C. H. Grant, C.-H. Hsieh, L. J. Richards, J. T. Schnute, and G. Sugihara, Equation-free mechanistic ecosystem forecasting using empirical dynamic modeling, Proc. Natl. Acad. Sci. USA **112**, E1569 (2015).
[198] G. Berkooz, P. Holmes, and J. L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annu. Rev. Fluid Mech. **25**, 539 (1993).
[199] D. Ruelle and F. Takens, On the nature of turbulence, Commun. Math. Phys. **20**, 167 (1971).
[200] C. P. Jackson, A finite-element study of the onset of vortex shedding in flow past variously shaped bodies, J. Fluid Mech. **182**, 23 (1987).
[201] Z. Zebib, Stability of viscous flow past a circular cylinder, J. Eng. Math. **21**, 155 (1987).
[202] T. Colonius and K. Taira, A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions, Comput. Methods Appl. Mech. Eng. **197**, 2131 (2008).
[203] See https://github.com/dynamicslab/pysindy.
[204] B. M. de Silva, K. Champion, M. Quade, J.-C. Loiseau, J. N. Kutz, and S. L. Brunton, PySINDy: A Python package for the sparse identification of nonlinear dynamics from data, J. Open Source Software **5**, 2104 (2020).

[205] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, Data-driven discovery of coordinates and governing equations, Proc. Natl. Acad. Sci. USA **116**, 22445 (2019).
[206] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, A. Ramadhan, and A. Edelman, Universal differential equations for scientific machine learning, arXiv:2001.04385.
[207] M. Gelbrecht, N. Boers, and J. Kurths, Neural partial differential equations for chaotic systems, New J. Phys. **23**, 043005 (2021).
[208] M. Kalia, S. L. Brunton, H. G. E. Meijer, C. Brune, and J. N. Kutz, Learning normal form autoencoders for data-driven discovery of universal, parameter-dependent governing equations, arXiv:2106.05102.
[209] M. Sorokina, S. Sygletos, and S. Turitsyn, Sparse identification for nonlinear optical communication systems: SINO method, Opt. Express **24**, 30433 (2016).
[210] M. Dam, M. Brøns, J. J. Rasmussen, V. Naulin, and J. S. Hesthaven, Sparse identification of a predator-prey system from simulation data of a convection model, Phys. Plasmas **24**, 022310 (2017).
[211] J.-C. Loiseau and S. L. Brunton, Constrained sparse Galerkin regression, J. Fluid Mech. **838**, 42 (2018).
[212] J.-C. Loiseau, B. R. Noack, and S. L. Brunton, Sparse reduced-order modeling: Sensor-based dynamics to full-state estimation, J. Fluid Mech. **844**, 459 (2018).
[213] L. Boninsegna, F. Nüske, and C. Clementi, Sparse learning of stochastic dynamical equations, J. Chem. Phys. **148**, 241723 (2018).
[214] P. Gelß, S. Klus, J. Eisert, and C. Schütte, Multidimensional approximation of nonlinear dynamical systems, J. Comput. Nonlinear Dyn. **14**, 061006 (2019).
[215] S. Thaler, L. Paehler, and N. A. Adams. Sparse identification of truncation errors, J. Comput. Phys. **397**, 108851 (2019).
[216] K. Kaheman, E. Kaiser, B. Strom, J. N. Kutz, and S. L. Brunton, Learning discrepancy models from experimental data, arXiv:1909.08574 v1.
[217] J.-C. Loiseau, Data-driven modeling of the chaotic thermal convection in an annular thermosyphon, Theor. Comput. Fluid Dyn. **34**, 339 (2020).
[218] S. Beetham and J. Capecelatro, Formulating turbulence closures using sparse regression with embedded form invariance, Phys. Rev. Fluids **5**, 084611 (2020).
[219] M. Schmelzer, R. P. Dwight, and P. Cinnella, Discovery of algebraic Reynolds-stress models using sparse symbolic regression, Flow, Flow, Turbul. Combust. **104**, 579 (2020).
[220] B. M. de Silva, D. M. Higdon, S. L. Brunton, and J. N. Kutz, Discovery of physics from data: Universal laws and discrepancies, Front. Artif. Intell. **3**, 25 (2020).
[221] J. J. Bramburger and J. N. Kutz, Poincaré maps for multiscale physics discovery and nonlinear Floquet theory, Physica D **408**, 132479 (2020).
[222] J. L. Callaham, J.-C. Loiseau, G. Rigas and S. L. Brunton, Nonlinear stochastic modelling with Langevin regression, Proc. R. Soc. London A **477**, 20210092 (2021).
[223] J. J. Bramburger, J. N. Kutz, and S. L. Brunton, Data-driven stabilization of periodic orbits, IEEE Access **9**, 43504 (2021).
[224] D. E. Shea, S. L. Brunton, and J. N. Kutz, SINDy-BVP: Sparse identification of nonlinear dynamics for boundary value problems, Phys. Rev. Res. **3**, 023255 (2021).
[225] S. Beetham, R. O. Fox, and J. Capecelatro, Sparse identification of multiphase turbulence closures for coupled fluid-particle flows, J. Fluid Mech. **914**, A11 (2021).
[226] A. A. Kaptanoglu, K. D. Morgan, C. J. Hansen, and S. L. Brunton, Physics-constrained, low-dimensional models for MHD: First-principles and data-driven approaches, Phys. Rev. E **104**, 015206 (2021).
[227] N. Deng, B. R. Noack, M. Morzyński, and L. R. Pastur, Galerkin force model for transient and post-transient dynamics of the fluidic pinball, J. Fluid Mech. **918**, A4 (2021).
[228] J. L. Callaham, S. L. Brunton, and J.-C. Loiseau, On the role of nonlinear correlations in reduced-order modeling, J. Fluid Mech. **938**, A1 (2022).
[229] J. L. Callaham, G. Rigas, J.-C. Loiseau, and S. L. Brunton, An empirical mean-field model of symmetry-breaking in a turbulent wake, Sci. Adv. **8**, eabm4786 (2022).
[230] C. Joshi, S. Ray, L. M. Lemma, M. Varghese, G. Sharp, Z. Dogic, A. Baskaran, and M. F. Hagan, Data-driven discovery of active nematic hydrodynamics, Phys. Rev. Lett. **129**, 258001 (2022).
[231] P. A. K. Reinbold, L. M. Kageorge, M. F. Schatz, and R. O. Grigoriev, Robust learning from noisy, incomplete, high-dimensional experimental data via physically constrained symbolic regression, Nat. Commun. **12**, 3219 (2021).
[232] E. P. Alves and F. Fiuza, Data-driven discovery of reduced plasma physics models from fully kinetic simulations, Phys. Rev. Res. **4**, 033192 (2022).
[233] H. Schaeffer and S. G. McCalla, Sparse model selection via integral terms, Phys. Rev. E **96**, 023302 (2017).
[234] D. A. Messenger and D. M. Bortz, Weak SINDy for partial differential equations, J. Comput. Phys. **443**, 110525 (2021).
[235] D. R. Gurevich, P. A. Reinbold, and R. O. Grigoriev, Robust and optimal sparse regression for nonlinear PDE models, Chaos **29**, 103113 (2019).
[236] P. A. Reinbold, D. R. Gurevich, and R. O. Grigoriev, Using noisy or incomplete data to discover models of spatiotemporal dynamics, Phys. Rev. E **101**, 010203(R) (2020).
[237] N. Joeman, M. Pradeep, L. K. Rajulapati, and R. Rengaswamy, Discovering governing partial differential equations from noisy data, Comp. & Chem. Eng. **180**, 108480 (2024).
[238] I. Abramovic, E.P. Alves, and M. Greenwald, Data-driven model discovery for plasma turbulence modelling, J. Plasma Phys. **88**, 895880604 (2022).
[239] S. Arbabi and M. Sahimi, The transition from Darcy to nonlinear flow in heterogeneous porous media: ISingle-phase flow, Transp. Porous Media (2024), doi:10.1007/s11242-024-02070-3.
[240] T. Qin, K. Wu, and D. Xiu, Data driven governing equations approximation using deep neural networks, J. Comput. Phys. **395**, 620 (2019).
[241] D. M. DiPietro, S. Xiong, and B. Zhu, Sparse symplectically integrated neural networks, in *Proceedings of the 34th Conference on Advances in Neural Information Processing Systems 33, Vancouver, Canada*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (NeurIPS, 2020).
[242] L. Lu, P. Jin, and G. E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nat. Mach. Intell. **3**, 218 (2021).

[243] B. Reyes, A. A. Howard, P. Perdikaris, and A. M. Tartakovsky, Learning unknown physics of non-Newtonian fluids, Phys. Rev. Fluids **6**, 073301 (2021).

[244] S. Cai, Z. Wang, L. Lu, T. A. Zaki, and G. E. Karniadakis, DeepM&Mnet: Inferring the electroconvection multiphysics fields based on operator approximation by neural networks, J. Comput. Phys. **436**, 110296 (2021).

[245] Z. Mao, L. Lu, O. Marxen, T. A. Zaki, and G. E. Karniadakis, DeepM&Mnet for hypersonics: Predicting the coupled flow and finite-rate chemistry behind a normal shock using neural-network approximation of operators, J. Comput. Phys. **447**, 110698 (2021).

[246] N. Shokri, A. Hassani, and M. Sahimi, Soil salinization, from pore to global scale: Mechanisms, modeling and outlook, Rev. Geophys. (to be published).

[247] A. Pershin, C. Beaume, K. Li, and S. M. Tobias, Training a neural network to predict dynamics it has never seen, Phys. Rev. E **107**, 014304 (2023).

[248] Y. Shin, J. Darbon, and G. E. Karniadakis, On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs, Commun. Comput. Phys. **28**, 2042 (2020).

[249] S. Mishra and R. Molinaro, Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs, IMA J. Numer. Anal. **42**, 981 (2022).

[250] T. De Ryck and S. Mishra, Error analysis for physics-informed neural networks (PINNs) approximating Kolmogorov PDEs, Adv. Comput. Math. **48**, 79 (2022).

[251] Y. Shin, Z. Zhang, and G. E. Karniadakis, Error estimates of residual minimization using neural networks for linear PDEs, J. Mach. Learn. Model. Comput. **4**, 73 (2023).

[252] H. Jasak, A. Jemcov, and Z. Tuković, OpenFOAM: A C^{++} library for complex physics simulations, in *Proceedings of the International Workshop on Coupled Methods in Numerical Dynamics, IUC Dubrovnik, Croatia, 2007*, edited by Z. Terze (University of Zagreb, Zagreb, Croatia, 2007).

[253] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rohnes, and G. N. Wells, The FEniCS project version 1.5, Arch. Numer. Softw. **3**, 9 (2015).

[254] W. D. Collins *et al.*, The formulation and atmospheric simulation of the community atmosphere model version 3 (CAM3), J. Clim. **19**, 2144 (2006).