Journal of Computational Physics 434 (2021) 110219

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Data-driven discovery of coarse-grained equations

Joseph Bakarji, Daniel M. Tartakovsky *

Department of Energy Resources Engineering, Stanford University, 367 Panama Mall, Stanford, CA 94305, USA

ARTICLE INFO

Article history:
Available online 19 February 2021

Keywords:
Machine learning
Closure approximation
Coarse-graining
Stochastic

ABSTRACT

Statistical (machine learning) tools for equation discovery require large amounts of data that are typically computer generated rather than experimentally observed. Multiscale modeling and stochastic simulations are two areas where learning on simulated data can lead to such discovery. In both, the data are generated with a reliable but impractical model, e.g., molecular dynamics simulations, while a model on the scale of interest is uncertain, requiring phenomenological constitutive relations and ad-hoc approximations. We replace the human discovery of such models, which typically involves spatial/stochastic averaging or coarse-graining, with a machine-learning strategy based on sparse regression that can be executed in two modes. The first, direct equation-learning, discovers a differential operator from the whole dictionary. The second, constrained equation-learning, discovers only those terms in the differential operator that need to be discovered, i.e., learns closure approximations. We illustrate our approach by learning a deterministic equation that governs the spatiotemporal evolution of the probability density function of a system state whose dynamics are described by a nonlinear partial differential equation with random inputs. A series of examples demonstrates the accuracy, robustness, and limitations of our approach to equation discovery.

1. Introduction

Empiricism, or the use of data to discern fundamental laws of nature, lies at the heart of the scientific method. With the advent of "machine learning", this ancient facet of pursuit of knowledge takes the form of inference, from observational or simulated data, of either analytical relations between inputs and outputs [24] or governing equations for system states [11,16,18,19,27,37]. The advantage of learning a governing equation, rather than an input-output map for a quantity of interest (QoI), is the possibility to "generalize" (extrapolate) over space and time and over different external inputs such as initial and boundary conditions. In this sense, learning a differential equation is akin to learning an iterative algorithm that generates a solution, rather than learning the solution itself. Of direct relevance to the present study is the use of sparse regression on noisy data to estimate the constant coefficients in nonlinear ordinary [8] and partial [22,23] differential equations (ODEs and PDEs, respectively). This strategy has been generalized to recover variable coefficients [21] or nonlinear constitutive relations between several state variables [28].

In physical sciences, observational data are seldom, if ever, sufficient to accomplish this goal; instead, the data must be generated by solving a governing equation. This strategy provides a partial explanation for why machine learning methods are yet to discover new physical laws: to generate the data one needs to know the underlying equation, which is subse-

* Corresponding author.
E-mail address: tartakovsky@stanford.edu (D.M. Tartakovsky).

https://doi.org/10.1016/j.jcp.2021.110219
0021-9991/

quently learned from these data. Multiscale modeling and stochastic simulations are two areas where learning on simulated data can lead to real discovery. In multiscale simulations, one is reasonably sure of an appropriate model at one scale (e.g., basic laws of molecular dynamics) and aims to learn a model at another scale (e.g., a continuum-scale PDE) from the data generated at the first scale. Examples of machine learning techniques for upscaling, i.e., discovery of coarse-grained dynamics from fine-grained simulations, and downscaling, i.e., discovery of fine-grained dynamics from coarse-grained simulations, can be found in [4,9,10,25].

In stochastic simulations, one deals with governing equations that either contain uncertain (random) parameters or are driven by randomly fluctuating forcings that represent sub-grid variability and processes (e.g., Langevin equations and fluctuating Navier-Stokes equations). Solutions of such problems, or QoIs derived from them, are given in terms of their probability density functions (PDFs). The goal here is to learn the deterministic dynamics of either the PDF of a system state (e.g., the Fokker-Planck equation for a given Langevin equation [20]) or its statistical moments (e.g., a PDE describing the spatiotemporal evolution of the ensemble mean of the system state [36]). Human (as opposed to machine) learning of such deterministic PDEs or their nonlocal (integro-differential) counterparts relies, e.g., on stochastic homogenization of the underlying stochastic models or on the method of distributions [29]. The latter provides a systematic way to derive deterministic PDF or CDF (cumulative distribution function) equations, regardless of whether the noise is white or colored [34]. Stochastic computation via the method of distributions can be orders of magnitude faster than high-resolution Monte Carlo [1,38].

While under certain conditions PDF equations can be exact, in general (e.g., when the noise is multiplicative and/or correlated) their derivation requires a closure approximation [1,33,38]. Such closures are usually derived either through perturbation expansions in the (small) variances of the input parameters or by employing heuristic arguments. Both approaches require considerable field-specific knowledge and can introduce uncontrollable errors. We propose to replace the human learning of PDF/CDF equations with a machine learning method to infer closure terms from data. It is based on sparse regression for discovering relevant terms in a differential equation [8,23,24], although its goals are different. The data come from a relatively few Monte Carlo runs of the underlying differential equation with random inputs, rather than from elusive observational data. Our approach amounts to coarse-graining in probability space and is equally applicable to deterministic coarse-graining as well.

We posit that sparse regression for PDE learning is better suited for PDF/CDF equations than for general PDEs. First, random errors in data and/or random fluctuations in an underlying physical process undermine the method's ability to learn a governing equation [21]; yet, their distributions might be easier to handle because of the smoothness of corresponding PDFs/CDFs [5,6]. Second, the known properties of distributions and PDF/CDF equations significantly constrain the dictionary of possible terms, rendering the equation learning more tractable and truly physics-informed. For example, a PDF equation has to be conservative (i.e., has to conserve probability); and, according to the Pawula theorem [20, pp. 63-95], the Kramers-Moyal expansion (i.e., a Taylor-series expansion of a master equation) should stop at the first three terms to preserve a PDF's positivity (giving rise to the Fokker-Plank equation). Finally, PDF equations tend to be linear, even if the underlying physical law describing each realization is nonlinear [29], which also limits the dictionary's size. Such considerations are, or should be, a key feature of *physics-informed* machine learning.

Our strategy to learn PDF equations from noisy data is presented in Section 2. A series of computational experiments in Section 3 is used to illustrate the robustness and accuracy of our approach. Main conclusions drawn from our study are summarized in Section 4.

2. Autonomous learning of PDF equations and their closure approximations

We start by formulating in Section 2.1 a generic problem described by a nonlinear PDE with uncertain (random) parameters and/or driving forces. A deterministic equation for the PDF of its solution is formulated in Section 2.2. In Section 2.3, we present two sparse-regression strategies for discovery of PDF equations. These are referred to as direct equation learning (DEL) and constrained equation learning (CEL).

2.1. Problem formulation

Consider a real-valued system state $u(\mathbf{x},t): D\times\mathbb{R}^+\to D_u$ that is defined on the d-dimensional spatial domain $D\subset\mathbb{R}^d$ and has a range $D_u\subset\mathbb{R}$. Its dynamics is described by a PDE

$$\frac{\partial u}{\partial t}+\mathcal{N}_{\mathbf{x}}(u;\boldsymbol{\lambda}_{\mathcal{N}})=g(u;\boldsymbol{\lambda}_g),\qquad \mathbf{x}\in D,\quad t>0, \tag{1}$$

which is subject to an initial condition $u(\mathbf{x},0)=u_{\text{in}}(\mathbf{x})$ and boundary conditions on the boundary ∂D of D. Our method applies to any combination of initial and boundary conditions for which the resulting initial/boundary value problem is well-posed; to be specific, we consider a Dirichlet condition $u(\mathbf{x},t)=u_{\text{b}}(\mathbf{x},t)$ for $\mathbf{x}\in\partial D$. The linear or nonlinear differential operator $\mathcal{N}_{\mathbf{x}}$ contains derivatives with respect to $\mathbf{x}$ and is parameterized by a set of coefficients $\boldsymbol{\lambda}_{\mathcal{N}}(\mathbf{x},t)$. The source term $g(u)$, a real-valued smooth function of its argument, involves another set of parameters $\boldsymbol{\lambda}_g(\mathbf{x},t)$. The system parameters $\boldsymbol{\lambda}=\{\boldsymbol{\lambda}_{\mathcal{N}},\boldsymbol{\lambda}_g\}$ are uncertain and treated as random fields. They are characterized by a single-point joint PDF $f_{\boldsymbol{\lambda}}(\boldsymbol{\Lambda};\mathbf{x},t)$ and a two-point covariance function (a matrix) $\mathbf{C}_{\boldsymbol{\lambda}}(\mathbf{x},t;\mathbf{y},\tau)$, both of which are either inferred from data or provided by expert

knowledge. The auxiliary functions $u_{\text{in}}(\mathbf{x})$ and $u_{\text{b}}(\mathbf{x},t)$ are also uncertain, being characterized by their respective single-point PDFs $f_{u_{\text{in}}}(U;\mathbf{x})$ and $f_{u_{\text{b}}}(U;\mathbf{x},t)$ and appropriate spatiotemporal auto-correlation functions.

Uncertainty in the input parameters renders predictions of the system state $u(\mathbf{x},t)$ uncertain (random) as well. Consequently, the full solution to (1) is infinitely-dimensional joint PDF of $u(\mathbf{x},t)$ at every space-time point $(\mathbf{x},t)$ or, if the space-time domain $D\times[0,T]$ is discretized into N_{dis} points, its N_{dis}-dimensional counterpart. Under certain conditions, a governing equation for such PDFs can be obtained by transforming (1) into a differential equation for the Hopf functional of $u(\mathbf{x},t)$ [32]. Solving the latter is nontrivial and computationally expensive. Instead, our goal is to compute the single-point PDF $f_u(U;\mathbf{x},t)$ of $u(\mathbf{x},t)$, whose mean $\mathbb{E}(u)\equiv\bar{u}(\mathbf{x},t)=\int U f_u(U;\mathbf{x},t)\mathrm{d}U$ and variance $\sigma_u^2(\mathbf{x},t)=\int U^2 f_u(U;\mathbf{x},t)\mathrm{d}U-\bar{u}^2$ (the integration is over D_u) serve as an unbiased prediction and a measure of predictive uncertainty, respectively.

Multiple uncertainty propagation tools can be used to estimate the PDF $f_u(U;\mathbf{x},t)$. These include (multilevel) Monte Carlo simulations (e.g., [30] and the references therein), which require one to draw multiple realizations of the inputs $\{\boldsymbol{\lambda},u_{\text{in}},u_{\text{b}}\}$ and solve (1) for each realization. This and other uncertainty quantification techniques are typically computationally expensive and provide little (if any) physical insight into either the expected (average) dynamics or the dynamics of the full PDF f_u. The method of distributions provides such an insight by yielding a deterministic PDE, which describes the spatiotemporal evolution of $f_u(U;\mathbf{x},t)$.

2.2. PDF equations

Regardless of whether the differential operator $\mathcal{N}_{\mathbf{x}}$ in (1) is linear or nonlinear, the PDF $f_u(U;\mathbf{x},t)$ satisfies (in general, approximately) a $(d+1)$-dimensional linear PDE [29]

$$\frac{\partial f_u}{\partial t}+\mathcal{L}_{\tilde{\mathbf{x}}}(f_u;\boldsymbol{\beta})=0,\qquad \tilde{\mathbf{x}}\equiv(\mathbf{x},U)\in D\times D_u,\quad t>0, \tag{2}$$

with a set of coefficients $\boldsymbol{\beta}(\tilde{\mathbf{x}},t)$. According to the Pawula theorem [20, pp. 63-95], the linear differential operator $\mathcal{L}_{\tilde{\mathbf{x}}}$ can include either first, second or infinite-order derivatives with respect to $\tilde{\mathbf{x}}$. Since solving infinite-order PDEs is not practical, we only consider second-order PDF equations (i.e., Fokker-Planck equations). Transition from (1) to (2) involves two steps: projection of the d-dimensional (linear or nonlinear) PDE (1) onto a $(d+1)$-dimensional manifold with the coordinate $\tilde{\mathbf{x}}$, and coarse-graining (stochastic averaging) of the resulting $(d+1)$-dimensional linear PDE with random inputs.[1] For first-order hyperbolic PDEs, this procedure can be exact when the system parameters $\boldsymbol{\lambda}$ are certain [1] and requires closure approximations otherwise [7]. It is always approximate when PDEs involved are parabolic [3] or elliptic [38], in which case the meta-parameters $\boldsymbol{\beta}$ might depend on the moments of the PDF f_u in a manner akin to the Boltzmann equation. Identification of the coefficients $\boldsymbol{\beta}(\tilde{\mathbf{x}},t)$, some of which might turn out to be 0, is tantamount to physics-informed learning of PDF equations.

When the system parameters $\boldsymbol{\lambda}$ are random constants—or when a space-time varying parameter, e.g., random field $\lambda(\mathbf{x})$, is represented via a truncated Karhunen-Loève expansion in terms of a finite number N_{KL} of random variables $\lambda_1,\ldots,\lambda_{N_{\text{KL}}}$—the PDF equation (2) is approximate, but an equation for the joint PDF $f_{u\boldsymbol{\lambda}}(U,\boldsymbol{\Lambda};\mathbf{x},t)$ of the inputs $\boldsymbol{\lambda}$ and the output u,

$$\frac{\partial f_{u\boldsymbol{\lambda}}}{\partial t}+\hat{\mathcal{L}}_{\tilde{\mathbf{x}}}(f_{u\boldsymbol{\lambda}};\hat{\boldsymbol{\beta}})=0,\qquad \tilde{\mathbf{x}}\equiv(\mathbf{x},U)\in D\times D_u,\quad t>0, \tag{3}$$

is exact [33]. Similar to (2), the differential operator $\hat{\mathcal{L}}_{\tilde{\mathbf{x}}}$ is linear and consists of up to second-order derivatives with respect to $\tilde{\mathbf{x}}$; its dependence on $\boldsymbol{\Lambda}$ is parametric, $\hat{\boldsymbol{\beta}}=\hat{\boldsymbol{\beta}}(\boldsymbol{\Lambda},\mathbf{x},t)$. Since the number of parameters in the set $\boldsymbol{\lambda}$ can be very large, one has to solve (3) for multiple values of $\boldsymbol{\Lambda}$, which is computationally expensive. A workable alternative is to compute a PDF equation (2) for the marginal $f_u(U;\mathbf{x},t)$ by integrating (3) over $\boldsymbol{\Lambda}$. In general, this procedure requires a closure [33].

2.3. Physics-informed dictionaries

Traditional data assimilation approaches for parameter identification, and deep learning strategies for PDE learning, rely on *a priori* knowledge of a dictionary of plausible terms in the differential operator. This is where a key advantage of learning the dynamics of $f_u(U;\mathbf{x},t)$ in (2), rather than the underlying dynamics of $u(\mathbf{x},t)$ in (1), manifests itself. Theoretical properties of PDF equations significantly constrain the membership in a dictionary, ensuring a faster and more accurate convergence to an optimal solution. We propose two strategies for discovering the PDF equation: DEL seeks to learn the full operator in (2), and CEL utilizes partial knowledge of the operator. This is illustrated in the following formulation of an optimization problem.

[1] When the system parameters $\boldsymbol{\lambda}$ vary in space and/or time, PDF equations are typically space-time nonlocal [2,14], i.e., integro-differential; in that case, the derivation of (2) requires an additional localization step [14,39].

Our goal is to discover the differential operator

$$\mathcal{L}_{\tilde{\mathbf{x}}} = \boldsymbol{\beta}(\tilde{\mathbf{x}}, t) \cdot \underbrace{\left(1, \frac{\partial}{\partial \tilde{x}_1}, \cdots, \frac{\partial}{\partial \tilde{x}_{d+1}}, \frac{\partial^2}{\partial \tilde{x}_1^2}, \frac{\partial^2}{\partial \tilde{x}_1 \partial \tilde{x}_2}, \cdots, \frac{\partial^2}{\partial \tilde{x}_{d+1}^2}, \cdots\right)^\top}_{\text{The dictionary } \mathcal{H} \text{ consisting of } Q \text{ members}}, \tag{4}$$

where $\boldsymbol{\beta}(\tilde{\mathbf{x}}, t) = (\beta_1, \ldots, \beta_Q)^\top \in \mathbb{R}^Q$ is the Q-dimensional vector of unknown (variable) coefficients. This is accomplished by minimizing the residual

$$\mathcal{R}(\boldsymbol{\beta}) = \frac{\partial \hat{f}_u}{\partial t} + \mathcal{L}_{\tilde{\mathbf{x}}}(\hat{f}_u; \boldsymbol{\beta}), \tag{5}$$

for all points $(\tilde{\mathbf{x}}, t)$ in the domain $D \times D_u \times [0, T]$. Here, $\hat{f}_u$ is a discrete sampling of f_u, typically given by observation data (see section 2.4 for details). Accordingly, the vector of optimal coefficients, $\check{\boldsymbol{\beta}}$, is found as a solution of the minimization problem

$$\check{\boldsymbol{\beta}}(U, \mathbf{x}, t) = \underset{\boldsymbol{\beta}(U,\mathbf{x},t)}{\operatorname{argmin}} \left\{ \int_0^T \int_{D \times D_u} \mathcal{R}^2(\boldsymbol{\beta}) \, \mathrm{d}\tilde{\mathbf{x}} \mathrm{d}t + \gamma \|\boldsymbol{\beta}\|_1^2 \right\}. \tag{6}$$

The L_1 norm, $\|\cdot\|_1$, is a regularization term that provides sparsification of the PDF equation, with γ serving as a hyper-parameter coefficient. Discovery of the full operator $\mathcal{L}_{\tilde{\mathbf{x}}}$, i.e., the solution of (4)–(6) is referred to as DEL.

The challenge in making the optimization problem (6) generalize to unseen space-time points is to identify a proper dictionary of derivatives in (4) that balances model complexity and predictability. On the one hand, a larger hypothesis class $\mathcal{H}$ (here, parametrized by Q coefficients $\beta_q(U, \mathbf{x}, t)$ with $q = 1, \ldots, Q$) has a higher chance of fitting the optimal operator $\mathcal{L}_{\tilde{\mathbf{x}}}$ that honors $\hat{f}_u$. It does so by minimizing the bias at the cost of a higher variance. On the other hand, a smaller dictionary $\mathcal{H}$ discards hypotheses with large variance, automatically filtering out noise and outliers that prevent the model from generalizing.

Both features are often used in coordination to nudge the regression problem in the right direction. For instance, having variable instead of constant coefficients in (6) significantly increases the power of the model to describe simulation data. At the same time, the L_1 regularization favors the parsimony of (i.e., fewer terms in) the operator $\mathcal{L}_{\tilde{\mathbf{x}}}$; making the resulting PDF equation more interpretable and easier to manipulate analytically.

The construction of the dictionary in (4) and, hence, of the residual $\mathcal{R}(\boldsymbol{\beta})$ is guided by the following considerations. First, if the random state variable $u(\mathbf{x}, t)$ is represented by a master equation, the Pawula theorem provides an exhaustive dictionary for PDF/CDF equations, i.e., specifies the form of $\mathcal{L}_{\tilde{\mathbf{x}}}$. It states that a truncated Taylor expansion of the master equation (i.e., the Kramers-Moyal expansion) must contain no higher than second-order derivatives for the function f_u to be interpretable as a probability density; otherwise, it can become negative. Consequently, if we restrict our discovery to local PDEs, i.e., ignore the possibility of f_u satisfying integro-differential equations or PDEs with fractional derivatives, then the dictionary containing first- and second-order derivatives in (4) is complete.

Second, the learned PDE for $f_u(U, \mathbf{x}, t)$ has to conserve probability, $\int f_u \mathrm{d}U = 1$ for all $(\mathbf{x}, t) \in D \times [0, T]$, i.e., the differential operator in (4) must be of the form $\bar{\mathcal{L}}_{\tilde{\mathbf{x}}} = \nabla_{\tilde{\mathbf{x}}} \cdot (\bar{\boldsymbol{\beta}} \nabla_{\tilde{\mathbf{x}}})$, where $\bar{\cdot}$ designates operators, and their coefficients, in the conservative form of the PDF equation. Accordingly, $\bar{\mathcal{L}}_{\tilde{\mathbf{x}}}(\cdot; \bar{\boldsymbol{\beta}})$ is a subset of its non-conservative counterpart $\mathcal{L}_{\tilde{\mathbf{x}}}(\cdot; \boldsymbol{\beta})$ in (2). The conservative form not only constrains the form of the operator, but also facilitates its numerical approximation. For example, a conservation law can be discretized using a finite volume scheme ensuring that the learned solution conserves probability.

Remark 1. For a particular initial condition, the solution of the minimization problem (6) could predict the differential operator $\mathcal{L}_{\tilde{\mathbf{x}}}$ containing the derivatives of order higher than two. However, we are only interested in learning PDF equations that generalize over arbitrary initial conditions, guaranteeing both positivity and conservation of probability independently of how these PDEs are solved after being discovered.

In a typical coarse-graining procedure, only a fraction of the terms in a PDF/CDF equation (i.e., in the dictionary $\mathcal{H}$) are unknown [29] and need to be learned from data. For example, an ensemble mean $\langle IO \rangle$ of two random fields, the model input I and (a derivative of) the model output O is written as $\langle IO \rangle = \langle I \rangle \langle O \rangle + \langle I' O' \rangle$, where the prime $'$ indicates zero-mean fluctuations about the respective means. The first term in this sum is a known term in a coarse-grained PDE, while the second requires a closure approximation, i.e., needs to be discovered. When applied to (1), the method of distributions [29] leads to an operator decomposition $\mathcal{L}_{\tilde{\mathbf{x}}} = \mathcal{K}_{\tilde{\mathbf{x}}} + \mathcal{C}_{\tilde{\mathbf{x}}}$, where $\mathcal{K}_{\tilde{\mathbf{x}}}$ is a known differential operator and the unknown operator $\mathcal{C}_{\tilde{\mathbf{x}}}$ contains the closure terms to be learned. With this decomposition, the discretized residual (5) takes the form

$$\mathcal{R}(\boldsymbol{\beta}) = \frac{\partial \hat{f}_u}{\partial t} + \mathcal{K}_{\tilde{\mathbf{x}}}(\hat{f}_u; \boldsymbol{\eta}) + \mathcal{C}_{\tilde{\mathbf{x}}}(\hat{f}_u; \boldsymbol{\beta}), \tag{7}$$

with known coefficients $\boldsymbol{\eta}$ and unknown coefficients $\boldsymbol{\beta}$, which are a subset of their counterparts in (5). Minimization of the residual (7) lies at the heart of CEL. We posit that CEL provides a proper framework for physics-informed equation discovery, in which physics guides the construction of the operator $\mathcal{K}_{\tilde{\mathbf{x}}}$ and observational/simulated data are used to infer the unknown closure operator $\mathcal{C}_{\tilde{\mathbf{x}}}$. In general, physical and mathematical properties of the differential equations one aims to learn constrain the dictionary $\mathcal{H}$. Depending on the problem, the scientific literature is full of versatile physical constraints that can and should improve equation discovery.

Remark 2. While generalization is what all human and machine learning aims to achieve, experience shows that the set over which a model generalizes is always bounded. That is why it is important to keep the human in the loop of discovering ever more generalizable and interpretable models. With that purpose in mind, while deep learning techniques are good at fitting nonlinear functions, learning equations by sparse regression provides a better collaborative framework between the scientist and the machine.

2.4. Numerical implementation

Let $\hat{\mathbf{f}}_u \in \mathbb{R}^{M\times N\times P}$, with entries $\hat{f}_u^{ijk} \equiv f_u(U_i, \mathbf{x}_j, t_k)$ for $i \in [1, M]$, $j \in [1, N]$ and $k \in [1, P]$, be a (numerical) solution of (2), at nodes of the discretized $U \in D_u$, $\mathbf{x} \in D$, and $t \in [0, T]$, such that $U_i = U_0 + i\Delta U$, $t_k = t_0 + k\Delta t$, and j is defined according to an appropriate indexing scheme in which $\mathbf{x}_j$ spans the entire d-dimensional grid. In practice, the residual $\mathcal{R}$ in (5) is computed numerically at finite collocation points in the domain $D_u \times D \times [0, T]$ by approximating the derivatives in (4) via finite differences, fast Fourier transforms, total variation regularized differentiation, etc. [8,23]. In the discretized version of (6), the residual $\mathcal{R}_{ijk}$ is a third-order tensor corresponding to the discretized domain on which the solution $\hat{f}_u^{ijk}$ is given, and the integrals become the sums over i, j and k:

$$\check{\boldsymbol{\beta}}^{ijk} = \underset{\boldsymbol{\beta}^{ijk}}{\operatorname{argmin}} \left\{ \frac{1}{MNP} \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{P} \mathcal{R}_{ijk}^2(\boldsymbol{\beta}^{ijk}) + \gamma \|\boldsymbol{\beta}\|_1^2 \right\}, \tag{8}$$

where $\boldsymbol{\beta}^{ijk} = \boldsymbol{\beta}(U_i, \mathbf{x}_j, t_k)$ is a third-order tensor.

Since the coefficients $\boldsymbol{\beta}(U, \mathbf{x}, t)$ are functions of $(d + 2)$ arguments, a numerical solution of the optimization problem in (6) might be prohibitively expensive. For example, a simulated annealing strategy (e.g., [4] and the references therein) calls for discretizing the coefficients $\boldsymbol{\beta}^{ijk}$ at the grid points $(U_i, \mathbf{x}_j, t_k)$ at which the solution $\hat{f}_u^{ijk}$ is defined and optimizing over $\boldsymbol{\beta}^{ijk}$. With Q features in the dictionary $\mathcal{H}$, this strategy yields $Q \times M \times N \times P$ unknown coefficients β_q^{ijk} and an optimization problem with complexity $\mathcal{O}(Q\,M^3)$, where typically $M \approx 10^3$. Solving such a high-dimensional problem requires adequate computational resources, e.g., multithreading on GPUs, proper memory allocation, etc. It can be implemented by stacking the minimization problems over all grid points in one large matrix, as done in [21] for learning parametric PDEs.

A more efficient approach is to represent the variable coefficients $\beta_q(U, \mathbf{x}, t)$ via a series of orthogonal polynomial basis functions (e.g., Chebyshev polynomials), $\psi_r(\cdot)$, such that

$$\beta_q(U, \mathbf{x}, t) = \sum_{r}^{R}\sum_{s}^{S}\sum_{w}^{W} \alpha_q^{rsw}\psi_r(U)\psi_s(\mathbf{x})\psi_w(t), \qquad q = 1, \ldots, Q, \tag{9}$$

where $\alpha_q^{rsw} \in \mathbb{R}$ denote the $d_{\text{pol}} = RSW$ coefficients in the polynomial representation of $\boldsymbol{\beta}_q$. With this approximation, the minimization problem (6) is solved over the unknown coefficients $\boldsymbol{\alpha}^{rsw} = (\alpha_1^{rsw}, \ldots, \alpha_Q^{rsw}) \in \mathbb{R}^Q$. For $d_{\text{coef}} = Q\,d_{\text{pol}}$ unknown coefficients β_q^{ijk}, the optimization dimension is of order $\mathcal{O}(Q\,R^3)$, where typically $R \lesssim 10$. This dimension is many orders of magnitude smaller than the brute force parametric optimization in [21], so that the resulting optimization problem can be solved on a personal computer.

Given the data matrix $\hat{\mathbf{f}}_u \in \mathbb{R}^{M\times N\times P}$ and its numerical derivatives with respect to U, $\mathbf{x}$ and t from the dictionary (4), we build the derivative feature matrix

$$\mathbf{F} = \begin{bmatrix} 1 & \partial_{x_1}\hat{f}_u^{111} & \cdots & \partial_{x_d}\hat{f}_u^{111} & \partial_U\hat{f}_u^{111} & \cdots & \partial_U^2\hat{f}_u^{111} \\ 1 & \partial_{x_1}\hat{f}_u^{211} & \cdots & \partial_{x_d}\hat{f}_u^{211} & \partial_U\hat{f}_u^{211} & \cdots & \partial_U^2\hat{f}_u^{211} \\ \vdots & \vdots & \vdots & \ddots & \vdots & & \\ 1 & \partial_{x_1}\hat{f}_u^{MNP} & \cdots & \partial_{x_d}\hat{f}_u^{MNP} & \partial_U\hat{f}_u^{MNP} & \cdots & \partial_U^2\hat{f}_u^{MNP} \end{bmatrix} \in \mathbb{R}^{d_{\text{dis}}\times Q}, \quad d_{\text{dis}} = MNP; \tag{10}$$

and its corresponding label vector (i.e., the known part of the PDF equation); e.g., based on the CEL formulation of the residual in (7),

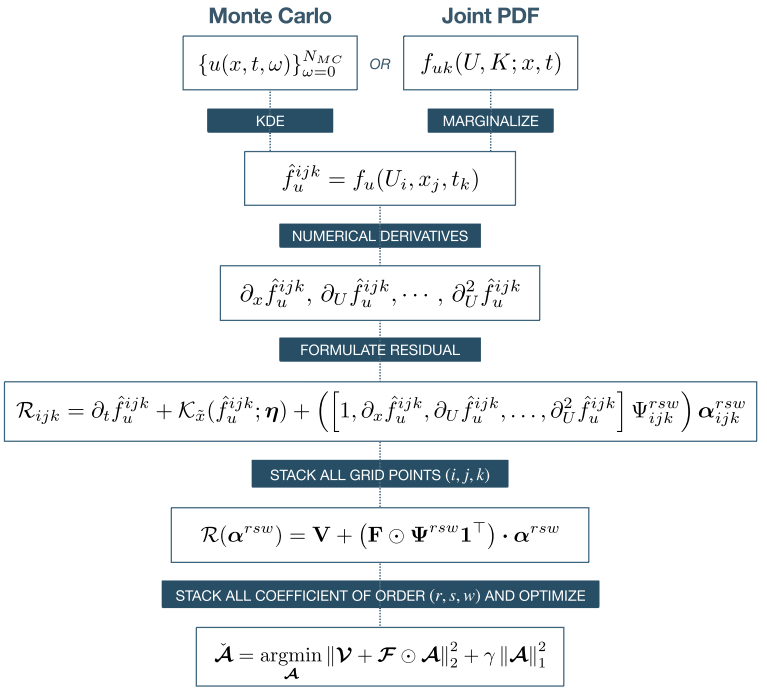

Fig. 1. A diagram of the algorithm for learning PDF equations from Monte Carlo simulations.

$$\mathbf{V} = \begin{bmatrix} \partial_t \hat{f}_u^{111} + \mathcal{K}_{\tilde{\mathbf{x}}}(\hat{f}_u^{111};\boldsymbol{\eta}) \\ \partial_t \hat{f}_u^{211} + \mathcal{K}_{\tilde{\mathbf{x}}}(\hat{f}_u^{211};\boldsymbol{\eta}) \\ \vdots \\ \partial_t \hat{f}_u^{MNP} + \mathcal{K}_{\tilde{\mathbf{x}}}(\hat{f}_u^{MNP};\boldsymbol{\eta}) \end{bmatrix} \in \mathbb{R}^{d_{\text{dis}}}. \tag{11}$$

For variable coefficients $\boldsymbol{\beta}(U,\mathbf{x},t)$, we define the vector $\boldsymbol{\Psi}^{rsw} \in \mathbb{R}^{d_{\text{dis}}}$ whose elements $\Psi_{ijk}^{rsw} \equiv \psi_r(U_i)\psi_s(\mathbf{x}_j)\psi_w(t_k)$ correspond to the grid-point elements in the columns of $\mathbf{F}$ and $\mathbf{V}$. For every polynomial coefficient vector $\boldsymbol{\Psi}^{rsw}$, the matrix form of the residual in (7) becomes

$$\mathcal{R}(\boldsymbol{\alpha}^{rsw}) = \mathbf{V} + (\mathbf{F} \odot \boldsymbol{\Psi}^{rsw}\mathbf{1}^\top)\boldsymbol{\alpha}^{rsw}, \tag{12}$$

where $\odot$ is the Hadamard (element-wise) product, $\mathbf{1} \in \mathbb{R}^Q$ is a vector of ones, such that the outer product $\boldsymbol{\Psi}^{rsw}\mathbf{1}^\top$ broadcasts the variable coefficient vector $\boldsymbol{\Psi}^{rsw}$ into Q identical columns. Let us introduce matrices

$$\boldsymbol{\mathcal{V}} = \begin{bmatrix} \mathbf{V} \\ \mathbf{V} \\ \vdots \\ \mathbf{V} \end{bmatrix} \in \mathbb{R}^{d_{\text{tot}}}, \qquad \boldsymbol{\mathcal{F}} = \begin{bmatrix} \mathbf{F} \odot \boldsymbol{\Psi}^{111}\mathbf{1}^\top \\ \mathbf{F} \odot \boldsymbol{\Psi}^{211}\mathbf{1}^\top \\ \vdots \\ \mathbf{F} \odot \boldsymbol{\Psi}^{RSW}\mathbf{1}^\top \end{bmatrix} \in \mathbb{R}^{d_{\text{tot}} \times Q}, \qquad \boldsymbol{\mathcal{A}} = \begin{bmatrix} \boldsymbol{\alpha}^{111} \\ \boldsymbol{\alpha}^{211} \\ \vdots \\ \boldsymbol{\alpha}^{RSW} \end{bmatrix} \in \mathbb{R}^{d_{\text{coef}}}, \tag{13}$$

where $d_{\text{tot}} = d_{\text{dis}} d_{\text{pol}}$. Then, minimization of the residual in (12) over all variable coefficients leads to the optimization problem

$$\check{\boldsymbol{\mathcal{A}}} = \underset{\boldsymbol{\mathcal{A}}}{\operatorname{argmin}} \|\boldsymbol{\mathcal{V}} + \boldsymbol{\mathcal{F}} \odot \boldsymbol{\mathcal{A}}\|_2^2 + \gamma \|\boldsymbol{\mathcal{A}}\|_1^2, \tag{14}$$

where $\|\cdot\|_2$ denoting the L_2 norm. A schematic representation of the resulting algorithm is shown in Fig. 1.

Following [8], our algorithm combines LASSO [31], i.e., L_1 regularization, with recursive feature elimination (RFE), which sequentially eliminates derivative features with small coefficients based on a tunable threshold at every iteration. This means that our algorithm has two hyper-parameters, γ and the RFE threshold, which are chosen based on the test set error (rather than being part of the optimization variable $\mathbf{A}$) and a desired sparsity (i.e., a variance-bias balance). For this purpose, we test a few cross-validation algorithms for parameter estimation from Python's `scikit-learn` package [17]. These algorithms, which rely on grid search to find the optimal regularization hyper-parameter γ, are `LassoCV` (an n-fold cross-validation set on each iteration), `LassoLarsCV` (an additional least-angle regression model), and `LassoLarsIC` (the Akaike or Bayes information criterion as an optimization variable over γ). They give very similar results when the optimal solution is in the

vicinity of the hypothesis class, but might differ significantly when the solution is far from optimal. In general, the choice of the algorithm depends on whether one favors more sparsity or accuracy.

For N_{MC} realizations of the random inputs, (1) is solved N_{MC} times on the discretized space-time domain $D \times [0, T]$, yielding N_{MC} solutions $u(\mathbf{x}, t)$. These Monte Carlo results are post-processed, e.g., with a Gaussian kernel density estimator (KDE) used in this study, to obtain the single-point PDF $f_u(U; \mathbf{x}, t)$ on the discretized domain $D_u \times D \times [0, T]$. The KDE bandwidth is estimated for every grid point in $D \times [0, T]$ using Scott's normal reference rule $h = 3.49\,\sigma\, N_{\text{MC}}^{-1/3}$ [26], where σ is the standard deviation of the data. The effect of the bandwidth on the solution is discussed in Appendix D. Both the kernel type and the bandwidth are added hyper-parameters that can be optimized.

The matrices $\boldsymbol{\mathcal{V}}$ and $\boldsymbol{\mathcal{F}}$ in (14) can be very large, depending on the selected order of the polynomials (R, S and W). We assume the coefficients to be time-independent, $\boldsymbol{\beta} = \boldsymbol{\beta}(U, \mathbf{x})$, so that $W = 1$. This makes the resulting optimization problems numerically tractable on a personal computer. To increase the computational efficiency, we exclude grid points on which the labels, e.g., $\partial_t f_u(U; \mathbf{x}, t)$, remain close to zero during the entire simulation. This sampling method leads to a significant reduction in computational cost (around a four-fold reduction in matrix size), especially in the case of a PDF that remains unchanged (equal zero) on the majority of the infinite domain.

To evaluate the generalization power of the method, we test its temporal extrapolation accuracy by fitting the hypothesis on the first 80% of the time horizon T, i.e., on the domain $\mathcal{D}_{\text{train}} = D_u \times D \times [0, 0.8T]$, and testing it on the remaining 20% of the simulation, i.e., on $\mathcal{D}_{\text{test}} = D_u \times \mathcal{D} \times (0.8T, T]$.

3. Results

We validate our approach on a set of nonlinear problems with uncertain initial conditions and parameters. In these experiments, we use the method of distributions [29] to map the PDE (1) for the random field $u(\mathbf{x}, t)$ onto either closed or unclosed PDEs of the marginal PDF $f_u(U; \mathbf{x}, t)$. This illustrates the difficulties associated with analytical derivation of a PDF/CDF equation, and shows how our data-driven approach to PDE discovery ameliorates them.

Section 3.1 deals with a nonlinear advection-reaction PDE driven by additive noise, for which the PDF equation is exact; this setting serves to validate our method by discovering a known PDF equation. In section 3.2, we consider a nonlinear advection-reaction PDE with multiplicative noise. The derivation of a corresponding PDF equation in this setting requires a closure approximation and, hence, our method leads to the discovery of a new PDF equation. In section 3.3, we analyze a nonlinear conservation law (Burgers' equation) subject to a random initial condition. This example demonstrates the importance of choosing an appropriate quantity, the single-point PDF or CDF of the state variable, for which to discover an equation.

3.1. Nonlinear advection-reaction PDE with additive noise

This experiment, in which the derivation of a PDF equation is exact, serves to test the method's accuracy in reconstruction of a PDF equation from N_{MC} Monte Carlo runs. Let $u(x, t)$ be a real-valued state variable, whose dynamics is governed by

$$\frac{\partial u}{\partial t} + k\frac{\partial u}{\partial x} = rg(u), \qquad x \in \mathbb{R}, \quad t \in \mathbb{R}^+ \tag{15}$$

where $k \in \mathbb{R}^+$ and $r \in \mathbb{R}^+$ are deterministic advection and reaction rate constant, respectively. The initial condition $u(x, 0) = u_0(x)$ is a random field with compact support in $\mathbb{R}$; it is characterized by a single-point PDF $f_{u_0}(U; x)$ and a two-point correlation function $\rho_{u_0}(x, y)$ specified for any two points $x, y \in \mathbb{R}$. The nonlinearity $g(u)$ is such that for any realization of $u_0(x)$ a solution of this problem, $u(x, t)$, is almost surely smooth. The PDF $f_u(U; x, t)$ satisfies exactly a PDE (Appendix A)

$$\frac{\partial f_u}{\partial t} + k\frac{\partial f_u}{\partial x} + r\frac{\partial g(U) f_u}{\partial U} = 0, \tag{16}$$

subject to the initial condition $f_u(U; x, 0) = f_{u_0}(U; x)$.

For the nonlinear source $g(u) = u^2$ used in this experiment, the analytical solution of (15) is $u(x, t) = [1/u_0(x - kt) - rt]^{-1}$. Uncertainty in the initial state, $u_0(x) = a \exp[-(x - \mu)^2/(2\sigma^2)]$, is incapsulated in the real constants a, μ, and σ. These parameters are sampled from independent Gaussian distributions, $a \sim \mathcal{N}(\eta_a, \xi_a)$, $\mu \sim \mathcal{N}(\eta_\mu, \xi_\mu)$, $\sigma \sim \mathcal{N}(\eta_\sigma, \xi_\sigma)$. The means and variances in these distributions are chosen to ensure that $u_0(x)$ almost surely has a compact support, $u(x \to \pm\infty, t) = 0$, which ensures integrability of $u(x, \cdot)$ on $\mathbb{R}$. We set $k = 1$, $r = 1$, $T = 0.5$, $\Delta t = 0.0085$, $x \in [-2.0, 3.0]$, $\Delta x = 0.0218$, $\Delta U = 0.0225$, $\eta_a = 0.8$, $\xi_a = 0.1$, $\eta_\mu = 0.5$, $\xi_\mu = 0.1$, $\eta_\sigma = 0.45$, $\xi_\sigma = 0.03$, and polynomial coefficients of order $M = 3$ and $N = 3$.

We use the grid search algorithm `LassoCV` to find $\gamma = 0.0004$ that minimizes the test-set error, while seeking a sparse solution tunable by the RFE threshold. This direct equation learning (DEL) procedure leads to a PDE,

$$\frac{\partial \hat{f}_u}{\partial t} + \mathbf{0.996}\,\frac{\partial \hat{f}_u}{\partial x} + \mathbf{0.955}\,U^2\frac{\partial \hat{f}_u}{\partial U} + \mathbf{2.06}\,U\frac{\partial \hat{f}_u}{\partial U} = 0, \tag{17}$$

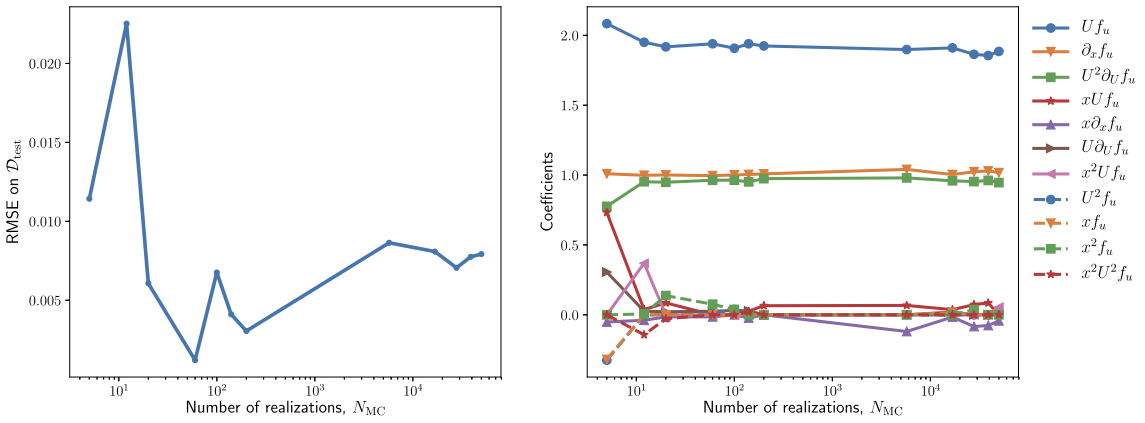

Fig. 2. Error in estimation of the PDF f_u on $\mathcal{D}_{\text{test}}$ (left) and the coefficients in the discovered PDF equation (17) (right) as function of the number of Monte Carlo realizations N_{MC}, without recursive feature elimination (RFE).

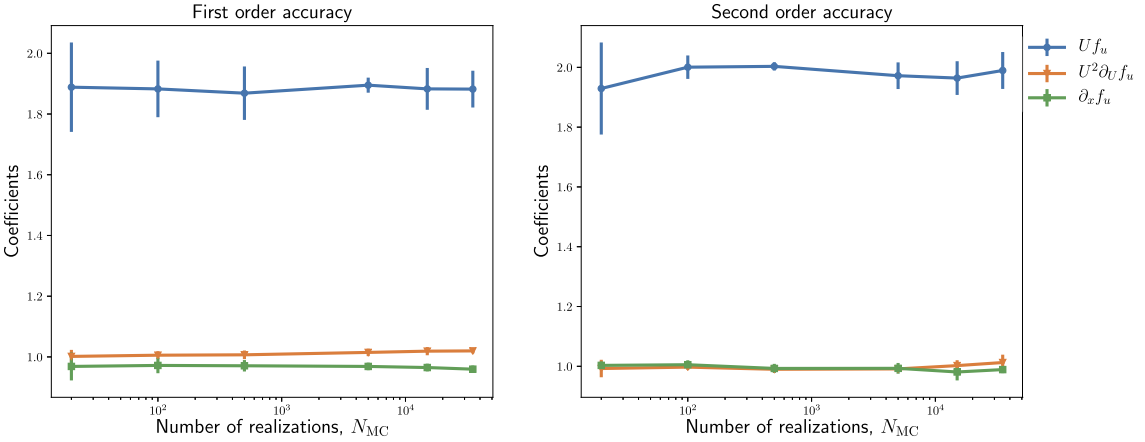

Fig. 3. The model coefficients, accompanied by error bars, estimated via first-order (forward difference) and second-order (central difference) accurate approximations of the derivatives. The second-order approximation leads to the smaller coefficient error and the comparable confidence intervals.

which demonstrates our method's ability to identify the relevant derivatives and their coefficients in equation (16) with $g(U) \equiv U^2$, eliminating all the remaining features in the dictionary $\mathcal{H}$; the original coefficients $k = 1$ and $r = 1$ are estimated with 3.2% error. In the absence of recursive feature elimination, the algorithm yields 11 non-zero terms (Fig. 2), highlighting the importance of using RFE sparsification in addition to L_1 regularization. This is due to the variance-bias trade-off discussed in section 2.3.

The amount and quality of simulated data are characterized by two hyper-parameters: the number of Monte Carlo runs, N_{MC}, and the mesh size, $\Delta = \max\{\Delta U, \Delta x, \Delta t\}$. Fig. 2 reveals that both the values of the coefficients $\boldsymbol{\beta}$ in the PDF equation (17) and the root mean square error (RMSE) of its solution $\hat{f}_u$ in the extrapolation mode are relatively insensitive to N_{MC} for around $N_{\text{MC}} > 20$ realizations. This means that, in this particular problem, the required number of Monte Carlo simulations is very small. But this is not always the case, as will be shown in section 3.3.

The average RMSE in Fig. 2 is on the order $\mathcal{O}(\Delta^2)$, where $\Delta \approx 0.02$. This error is equivalent to a numerical scheme's approximation error (a truncation error of the relevant Taylor expansion). The second-order error observed here could be due to the use of a first-order finite difference scheme to create the derivative features. In Fig. 3, we verify this hypothesis by comparing the coefficients $\boldsymbol{\beta}$ in (17) predicted via the first-order (forward difference) and second-order (central difference) approximations of the derivatives. The second-order approximation predicts the coefficients $(0.998, 0.994, 1.976)$ and, indeed, leads to a smaller error of 0.5%—relative to the first-order approximation error of 3.2%—with comparable confidence intervals.

A solution $u(x,t)$ to (15) can be (nearly) deterministic in a part of the space-time domain $\mathcal{S} \in D \times [0,T]$, e.g., when $u(x,t)$ has a compact support; in this experiment the size of $\mathcal{S}$ is controlled by the support of the initial state $u_0(x)$ which is advected by (15) throughout the space-time domain $\mathbb{R} \times [0,T]$. This situation complicates the implementation of KDE and numerical differentiation, because the resulting PDF $f_u(U;x,t)$ is (close) to the Dirac delta function $\delta(\cdot)$; in this experiment, $f_u(U;x,t) \sim \delta(U)$ for $(x,t) \in \mathcal{S}$, as shown in Fig. 4 for space-time points $(x = 2.03, t)$ with small t. Consequently, a numerical implementation of our algorithm must provide an adequate approximation of the delta function and be able to handle

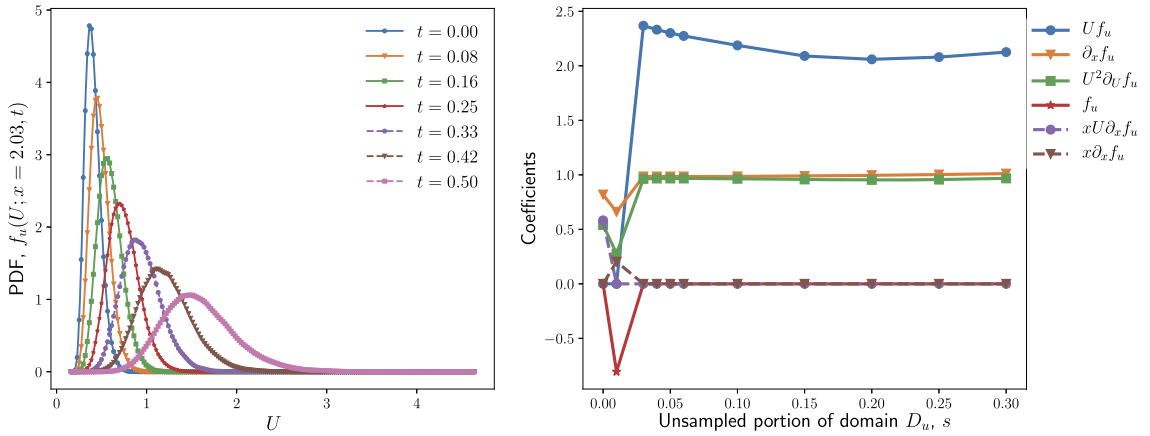

Fig. 4. PDF $f_u(U; x = 2.03, t)$ (left) and the coefficient values in the discovered PDF equation using DEL (right) The effect of omitting training samples from the U domain $D_u^o \in [0, s|D_u|]$, with $s \in [0, 1]$, where the sharp PDF profiles complicate numerical differentiation. An RFE threshold of 0.1 is used.

sharp gradients with respect to U in the neighborhood of $U = 0$. (We found that rejecting data points near $u(x, t) = 0$ from KDE leads to a poor MC approximation of $f_u(U; \cdot)$ and its derivatives, and to the discovery of an incorrect PDF equations on $\mathcal{D}_{\text{train}}$.) We address this issue by adding small perturbations ξ to the initial state $u_0(x)$, i.e., by generating the training data from (15) subject to the initial condition $u_0^{\text{m}}(x) = \xi + u_0(x)$, where the random variable ξ has the exponential PDF, $f_\xi(s) = \lambda \exp(-\lambda s)$ for $s \geq 0$ and $= 0$ for $s < 0$, with $\lambda \gg 1$ (in our experiments, $\lambda = 10$).[2] Another alternative is to omit training data from the simulation domain where the PDF has sharp profiles. In this case, the data in the domain $D_u^o \in [0, s|D_u|]$, with $s \in [0, 1]$, are excluded from the training set $\mathcal{D}_{\text{train}}$ (Fig. 4b). Other strategies, which we defer for follow-up studies, include the discovery of PDF/CDF equations in the frequency domain.

3.2. Nonlinear advection-reaction PDE with multiplicative noise

This test deals with a situation in which the exact PDF equation is not available and, hence, its multiple approximations have the right to exist, one of which we aim to discover. The setting is the same as in the previous section, except for the system parameter k in (15) that is now random rather than deterministic. It is statistically independent from the random initial state $u_0(x)$ and has the PDF $f_k(K)$ with ensemble mean $\langle k \rangle$ and standard deviation σ_k. In the simulations reported below, we take $f_k(K)$ to be Gaussian with $\langle k \rangle = 1$ and $\sigma_k = 0.2$.

The PDF $f_u(U; x, t)$ of the solution to this problem, $u(x, t)$, satisfies a PDE (Appendix A),

$$\frac{\partial f_u}{\partial t} + \langle k \rangle \frac{\partial f_u}{\partial x} + r \frac{\partial g(U) f_u}{\partial U} + \mathcal{C}(f_u) = 0. \tag{18}$$

This equation is formally exact but not computable since the operator $\mathcal{C}(f_u)$ is unknown. It has to be conservative and is generally nonlocal [14,15], taking the form of an integro-differential operator or, equivalently, a fractional-derivatives operator. One of its plausible approximations is (Appendix C)

$$\mathcal{C}(f_u) = -\sigma_k^2 \frac{\partial}{\partial x} \int_0^t \int_D \int_{D_u} G(U, V; x, y; t - \tau) \frac{\partial f_u}{\partial y}(V; y, \tau) \mathrm{d}V \mathrm{d}y \mathrm{d}\tau, \tag{19}$$

where the kernel G is the Green's function of the two-dimensional advection equation. The spatiotemporal localization of (19) yields a PDF equation of the form (2). While one could include nonlocal terms in the dictionary $\mathcal{H}$, we leave this endeavor for future work; instead, our goal is to learn the localized version of $\mathcal{C}(f_u)$ in (18). The resulting PDF equation, discovered via CEL (7), is compared with its counterpart discovered via DEL (5).

Fig. 5 exhibits the coefficients of the PDF equations discovered with these two alternative strategies for equation discovery. These equations are not, and do not have to be, identical because a closure approximation is not unique. Nevertheless, the overall structure of the differential operators identified with the two strategies is largely consistent: the first eight most relevant features identified by direct learning, and two most relevant features in the learning with decomposition, involve the terms (f_u, $\partial_x f_u$ and $\partial_U f_u$) that must be present in the PDF equation based on the theoretical considerations leading to the unclosed PDE (18). The next most relevant term identified by both learning strategies is around $-0.02\partial_x^2 f_u$, regardless

[2] The choice of an exponential distribution ensures that $f_u(U; x, t) = 0$ for $U < 0$, thus honoring the physical meaning of the random variable $u(x, t)$, e.g., solute concentration, that must stay positive throughout the simulation.

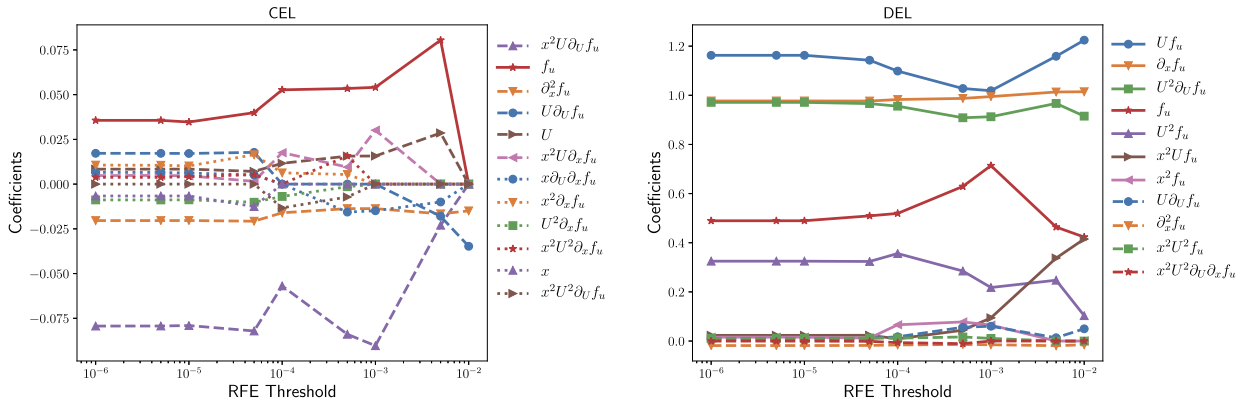

Fig. 5. The coefficients in the PDF equations, as function of the recursive feature elimination threshold, alternatively discovered via constrained equation learning (left), as in (7) or (18), and direct equation learning (right), as in (5).

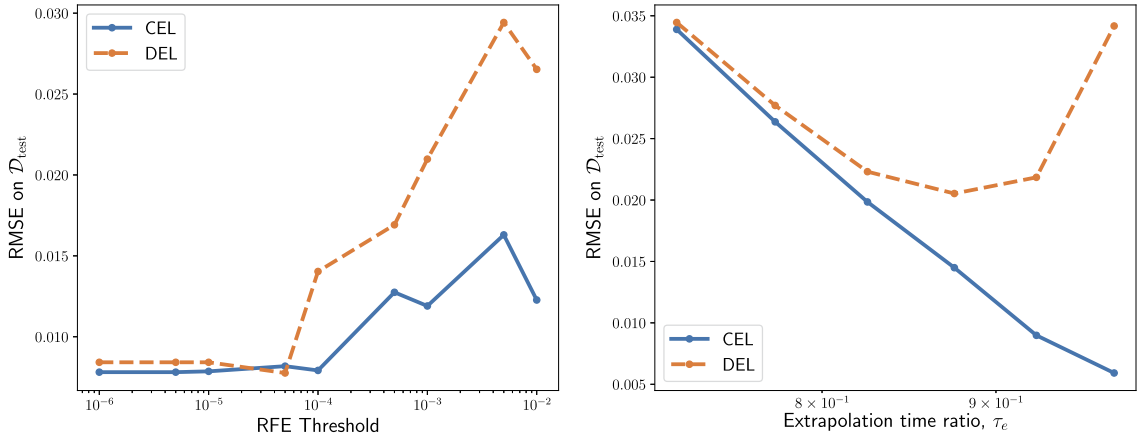

Fig. 6. Cross-validation error as function of the recursive feature elimination threshold for DEL (5) and CEL (18) (left). Cross-validation error as function of the extrapolation time fraction $\tau_e = t/T$ on the test set time domain $[0.7T, T]$, with corresponding training time domain $[0, 0.7T]$ (right). The RMSE of the latter is lower because it is constrained by known derivative features, which reduces the hypothesis set.

of the RFE threshold. Considering the value of $\sigma_k = 0.2$, this is consistent with a theoretical diffusion-type localized closure $\mathcal{C}(f_u) \approx -(\sigma_k^2/2)\partial_x^2 f_u$ in (19).

Fig. 5 reveals that the two learning strategies do differ in their identification of the variable coefficients $\boldsymbol{\beta}(U, x)$. For example, when small RFE thresholds are used, the direct learning identifies the coefficient of the feature f_u to be $0.5 + 1.2U + 0.35U^2$, instead of its counterpart $2U$ identified by the learning with decomposition. This discrepancy might either be a manifestation of non-uniqueness or reflect numerical differentiation error. Large RFE thresholds introduce additional dependency of this coefficient, x^2U.

Over a wide range of the RFE threshold, learning with decomposition (18) outperforms direct learning (5) in terms of the RMSE on the test set (Fig. 6). This proves the point stated in section 2.3: constraining the hypothesis class with physics improves the approximation accuracy. That is, the more terms in a PDE we know, the more accurate the discovered PDE is likely to be.

Remark 3. Given the lack of uniqueness, generalizability of a discovered PDE depends on the amount of information about the inputs one can transfer into its discovery. The PDF equation (18) with the closure $\mathcal{C}(f_u) \approx -(\sigma_k^2/2)\partial_x^2 f_u$ depends only on the first two moments of the random parameter k, its mean $\langle k \rangle$ and standard deviation σ_k, rather than on its full PDF $f_k(K)$. Hence, it might not generalize well to an arbitrary distribution of k. This can be remedied by learning a PDE for $f_{uk}(U, K; x, t)$, the joint PDF of the input k and output $u(x, t)$. We show in Appendix B that the latter satisfies exactly an initial value problem

$$\frac{\partial f_{uk}}{\partial t} + K\frac{\partial f_{uk}}{\partial x} + r\frac{\partial g(U) f_{uk}}{\partial x} = 0, \qquad f_{uk}(U, K; x, 0) = f_{u_0}(U; x) f_k(K), \tag{20}$$

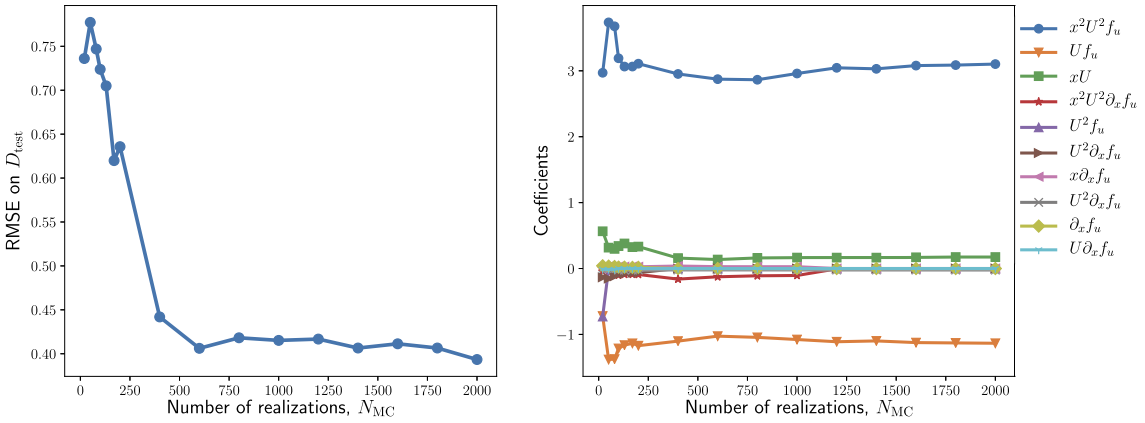

Fig. 7. Error in estimation of the PDF f_u on $\mathcal{D}_{\text{test}}$ (left) and the coefficients in the discovered PDF equation (right) for inviscid Burgers model without a shock. These are plotted as function of the number of Monte Carlo realizations N_{MC}, with the RFE threshold of 0.001. Only the features whose values exceed 0.01 are shown, out of 21 non-zero features.

for any given input parameter distribution $f_k(K)$. This suggests that marginalization of f_{uk} over k, i.e., transition from (20) to (18), transfers the information about $f_k(K)$ from the initial condition for the former to the differential operator in the latter; thus, leading to a weaker generalizability.

3.3. Nonlinear conservation laws

As a more nonlinear example of (1), we consider a smooth state variable[3] $u(x,t)$ whose dynamics is governed by a hyperbolic conservation law with a nonlinear flux $g(u)$,

$$\frac{\partial u}{\partial t} + \frac{\partial g(u)}{\partial x} = 0, \qquad u(x,0) = u_0(x), \qquad x \in \mathbb{R}, \quad t > 0. \tag{21}$$

Randomness comes from the uncertain initial state u_0, which is characterized by a single-point PDF $f_{u_0}(U;x)$ and a two-point correlation function $\rho_{u_0}(x,y)$. In the simulations reported below, we set $g(u) \equiv u^2/2$, i.e., analyze the inviscid Burgers equation; and $u_0(x) \equiv \xi + a\exp[-(x-b)^2/(2c^2)]$ with Gaussian variables $\xi \sim \mathcal{N}(\eta_\xi, \sigma_\xi)$, $a \sim \mathcal{N}(\eta_a, \sigma_a)$, $b \sim \mathcal{N}(\eta_b, \sigma_b)$, and $c \sim \mathcal{N}(\eta_c, \sigma_c)$. We focus on the simulation time horizon T before a shock develops. Once again, our goal is to discover a governing equation for $f_u(U;x,t)$, the single-point PDF of $u(x,t)$, from N_{MC} Monte Carlo realizations of (21).

When applied to (21) with smooth solutions, the method of distributions yields the exact integro-differential equation [29]

$$\frac{\partial f_u}{\partial t} + g'(U)\frac{\partial f_u}{\partial x} + g''(U)\frac{\partial F_u}{\partial x} = 0, \qquad F_u(U;x,t) = \int_{-\infty}^{U} f_u(\tilde{U};x,t)\mathrm{d}\tilde{U}, \tag{22}$$

where $g'(U)$ and $g''(U)$ are the first and second derivatives of $g(U)$, respectively. Its local approximation within the dictionary $\mathcal{H}$ in (4) is not unique. Given the superior behavior of CEL (7) observed above, we report in Fig. 7 the results of this strategy only. The RMSE of predicted PDF f_u over the test set $\mathcal{D}_{\text{test}}$ is an order of magnitude higher than in the previous experiments on advection-reaction equations. Both this error and the concomitant estimates of the coefficients in the learned PDF equation require a larger number of Monte Carlo realizations to stabilize compared to the previous experiment, and that with a significant error.

An alternative to learning a PDF equation for $f_u(U;x,t)$ is to discover a CDF equation that governs the dynamics of $F_u(U;x,t)$, the single-point CDF of $u(x,t)$. The integration of (23) over U leads to the PDE [29]

$$\frac{\partial F_u}{\partial t} + U\frac{\partial F_u}{\partial x} = 0, \tag{23}$$

which is exact. Both the uniqueness of this CDF equation and the form of its differential operator suggest that its discovery from the dictionary $\mathcal{H}$ in (4) might be more tractable than the discovery of the corresponding PDF equation. Fig. 8 demonstrates this to be the case. In the absence of the recursive feature elimination, the test set error is optimal when a few extra terms are included. These are advection ($\partial_x F_u$) and diffusion ($\partial_x^2 F_u$) terms that account for Monte Carlo sampling

[3] Strategies for handling discontinuities and shocks are discussed in the end of this section.

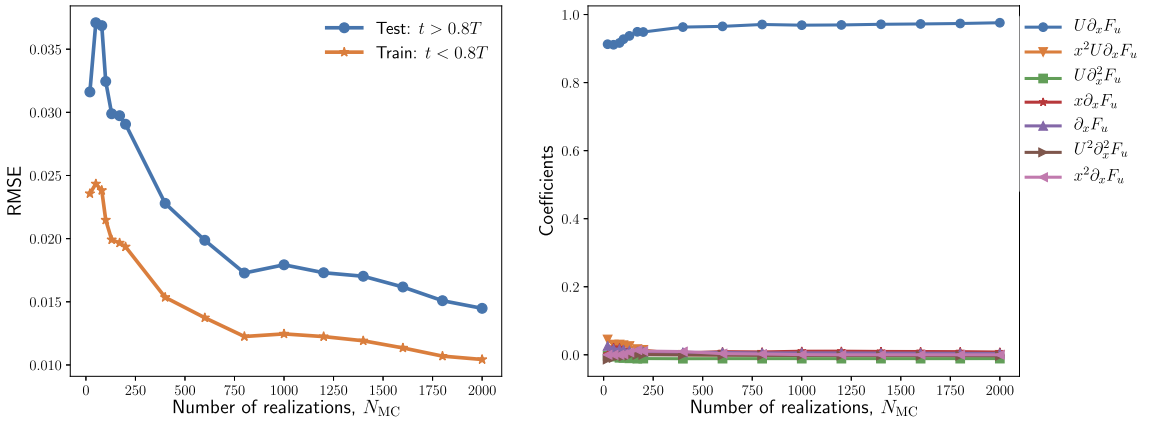

Fig. 8. Error in estimation of the CDF F_u on $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ (left) and the coefficients in the discovered CDF equation (right) for inviscid Burgers model without a shock. These are plotted as function of the number of Monte Carlo realizations N_{MC}. An RFE threshold of 0.001 is used and only the features whose values exceed 0.01 are shown, out of 22 non-zero features.

error and its associated KDE approximation. When an RFE threshold of 0.03 is used, with $N_{\text{MC}} = 50000$ realizations, the CDF equation is recovered almost exactly; with coefficient $1.001U$. This finding demonstrates the importance of choosing the "right" observable to learn. In this sense, it is akin to construction of reduced order models via the Koopman operator and dynamic mode decomposition [12,13].

The results presented above are for smooth solutions on the space-time domain before a shock or discontinuity develops. However, many nonlinear hyperbolic conservation laws (21), including the inviscid Burgers equation (21) with $g(u) \equiv u^2/2$, develop such discontinuities in finite time. When a shock develops, the PDE (21) is valid only in parts of the simulation domain where $u(x,t)$ is differentiable, and has to be supplemented with an equation for the position of and jump across the shock, i.e., with the Rankine-Hugoniot condition. Accordingly, PDF/CDF equations, such as (22) and (23), for problems with a shock are also invalid across the shock front and require special treatment [1,35].

Specifically, (23) has to be replaced with [6]

$$\frac{\partial F_u}{\partial t} + U\frac{\partial F_u}{\partial x} = \beta_1(U,x,t). \tag{24}$$

This CDF equation is exact even in the presence of shocks; while the "kinetic defect" $\beta_1(U,x,t)$ is unique, its functional form is generally unknown. This equation falls within the dictionary $\mathcal{H}$ in (4) and, hence, has a chance of being discovered from N_{MC} Monte Carlo realizations, $u^{(m)}(x,t)$ with $m = 1,\dots,N_{\text{MC}}$, of a solution to (21). These realizations are computed using a finite volume algorithm with a van Leer flux limiter to resolve the shock front without tracking its position. In this simulation, a periodic boundary condition is used.

As before, we explore two alternative strategies for equation discovery: DEL with the dictionary $\mathcal{H}$ in (4), and CEL in which only the source term $\beta_1(U,x,t)$ in (24) has to be learned. In both cases, the coefficients $\boldsymbol{\beta}$ are approximated with polynomials (9) in U and x; their time dependence is reflected through the selection of training data. Let $t_{\text{sh}}^{(m)}$ with $m = 1,\dots,N_{\text{MC}}$ denote the shock breaking times in each of the N_{MC} Monte Carlo realizations of (21). The test and training set sampling time interval $[t_{\text{st}}, t_{\text{end}}]$ is chosen to contain the minimal shock breaking time $t_{\text{sh}} \equiv \min\{t_{\text{sh}}^{(1)},\dots,t_{\text{sh}}^{(N_{\text{MC}})}\}$, such that $0 \le t_{\text{st}} \le t_{\text{sh}} \le t_{\text{end}} \le T$. We investigate the relative importance of pre- and post-shock data in terms of the post-shock sampling fraction,

$$p_s = \frac{t_{\text{end}} - t_{\text{sh}}}{t_{\text{end}} - t_{\text{st}}}. \tag{25}$$

Fig. 9 demonstrates the performance of the two learning strategies, DEL and CEL, as function of $p_s = t_{\text{end}}/t_{\text{sh}} - 1$; the latter corresponds to the sampling time interval whose length is kept constant, $|t_{\text{end}} - t_{\text{st}}| = t_{\text{sh}}$, for all p_s. The results of the DEL approach (top row) show that the learning algorithm accounts for the shock by adding advection terms in both U and x but with an increasing RMS error as the sampling time domain enters the post-shock region $t > t_{\text{sh}}$. Compared to the DEL approach, CEL with only polynomial features (bottom row) starts with a smaller cross-validation error for $p_s < 0.5$, showing the approximation power of this constrained approach, as justified by known existence theorems. The drastic increase in error for $p_s > 0.5$ can be explained by the fact that i) the optimal source term $\beta_1(U,x,t)$ is typically time-dependent (here assumed time-independent) and ii) a polynomial approximation of β_1 cannot capture a spatially localized shock region.

The similar trend in the change of the coefficients in CEL motivates the time-dependent representation of the kinetic defect $\beta_1(U,x,t) = \beta_1(U,x)\mathcal{I}(t \in [t_{\text{i}}, t_{\text{i+1}}])$, where $\mathcal{I}$ is the indicator function for the membership in disjoint sampling intervals $[t_{\text{i}}, t_{\text{i+1}}]$ on a regular grid, such that $i = 0,\dots,n-1$ and $n(t_{\text{i+1}} - t_{\text{i}}) < T$. While this approach is computationally more expensive than including time polynomials in the source term, it does not assume a specific functional form. We will investigate this and other strategies for explicit treatment of time dependence in a follow-up study.

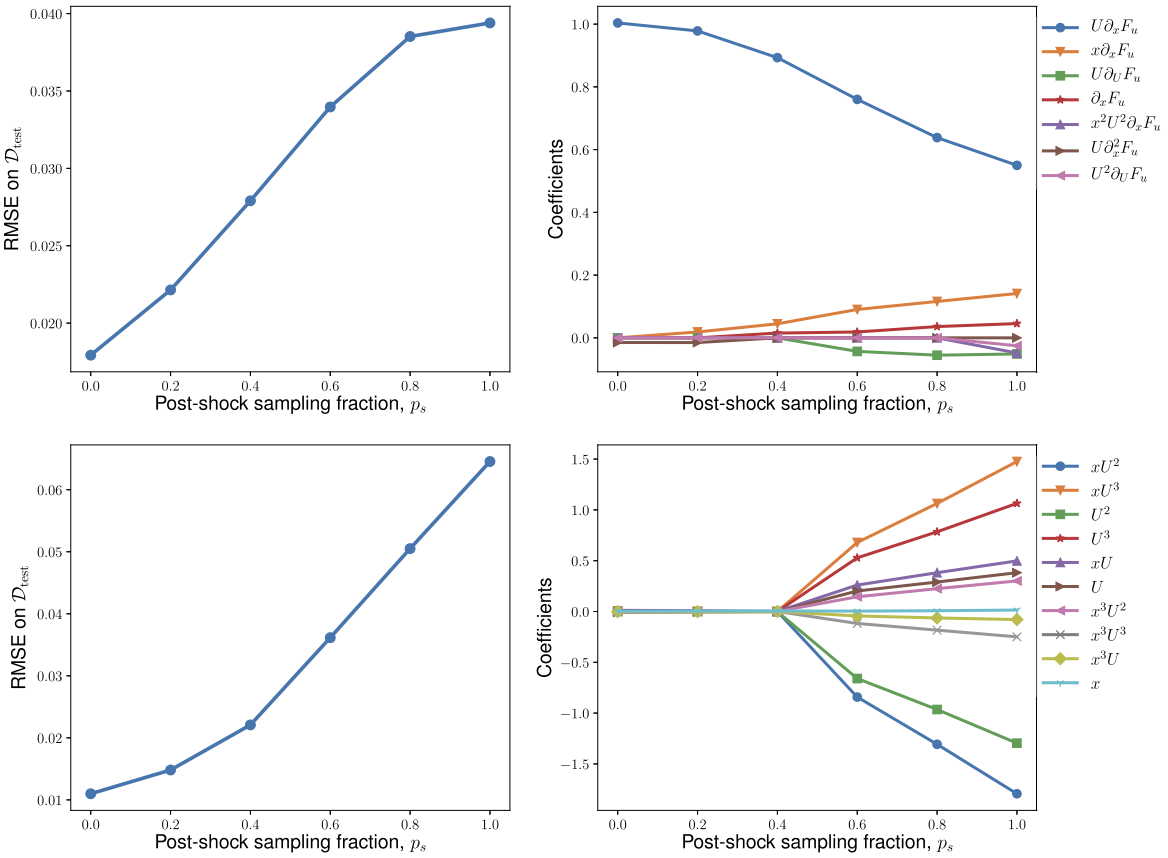

Fig. 9. Error in estimation of the CDF F_u on $\mathcal{D}_{\text{test}}$ (left column) and the learned coefficients (right column) for the inviscid Burgers model with a shock. These are plotted as function of the post-shock sampling fraction, p_s, defined in (25). DEL identifies the differential operator for the CDF equation (top row), while CEL infers the polynomial representation of the kinetic defect term (bottom row). Only the features whose values exceed 0.01 are shown, out of 23 non-zero features in the top row, and 27 features in the bottom row.

4. Discussion and conclusions

We presented a sparse-regression strategy for discovering coarse-grained equations from data generated with either a fine-scale model or Monte Carlo simulations of a model with random/uncertain coefficients and inputs. Motivated by the latter setting, we used this strategy to learn deterministic partial differential equations (PDEs) for the probability density function (PDF) or cumulative distribution function (CDF) of a random system state. The learning is not only data-driven but also physics-informed, in the sense that the construction of a dictionary of plausible terms in the differential operator, i.e., the formulation of scientific hypotheses, is driven by theoretical considerations such as the Pawula theorem [20, pp. 63-95]. Our sparse-regression strategy can be implemented in two modes. The first, direct equation-learning (DEL), discovers a differential operator from the whole dictionary. The second, constrained equation learning (CEL), discovers only those terms in the differential operator that need to be discovered, i.e., learns closure approximations.

Our analysis leads to the following major conclusions.

- Discovery of PDF/CDF equations is advantageous because they are known to be linear and to satisfy a number of theoretical constraints that reduce both the hypothesis set and the dictionary size.
- Selection of an observable whose dynamics is to be learned is key for successful equation discovery. In our example, the discovery of a CDF equation turned out to be significantly more robust than that of the corresponding PDF equation.
- The hyper-parameters used to fine-tune the algorithm, especially those used to generate and postprocess Monte Carlo data, play an important role in its performance. For example, the KDE bandwidth that leads to the accurate discovery of a differential operator can also add irrelevant terms.
- Our algorithm can be used to reverse-engineer black-box simulators by rediscovering the equations they solve from their output data. This can be used for solution verification, particularly when the solver does not strictly use rigorous numerical techniques, as occurs in physics-informed neural networks [18].

Future studies will deal with a number of theoretical and computational aspects of equation discovery, some of which are mentioned below. First, our framework provides a venue for hypothesis testing, since it relies on physical considerations to construct a dictionary of plausible terms in the operator. In this study, we constrained such a dictionary to the class of local models. Future studies will account for nonlocality by incorporating integro-differential or fractional-derivative terms in the dictionary.

Second, the accuracy and computational efficiency of the proposed algorithm require further investigation and improvement. Future lines of research include the deployment of advanced techniques for optimization over tunable hyperparameters such as the KDE bandwidth and the regularization coefficient.

Finally, we used a threshold on the labels $\partial_t f_u$ to exclude grid points on which the PDF does not change throughout the simulation from the training set. A more robust strategy might use the Kullback-Leibler divergence for feature elimination before running the optimization algorithm.

CRediT authorship contribution statement

Joseph Bakarji: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft. **Daniel M. Tartakovsky:** Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by Air Force Office of Scientific Research under award numbers FA9550-17-1-0417 and FA9550-18-1-0474, and by a gift from Total.

Appendix A. Derivation of the PDF equation

Consider a generalized function

$$\pi_u(U-u) \equiv \delta(U-u(x,t)), \tag{A.1}$$

where $\delta(\cdot)$ is the Dirac delta function. If the random variable u at any space-time point (x,t) has a PDF $f_u(U;x,t)$, then, by definition of the ensemble average $\mathbb{E}[\cdot]$,

$$\begin{aligned}\mathbb{E}[\pi_u(U-u)] &= \int_{-\infty}^{+\infty} \pi_u(U-\tilde{U}) f_u(\tilde{U};x,t)\mathrm{d}\tilde{U} \\ &= \int_{-\infty}^{+\infty} \delta(U-\tilde{U}) f_u(\tilde{U};x,t)\mathrm{d}\tilde{U} \\ &= f_u(U;x,t).\end{aligned} \tag{A.2}$$

That is, the ensemble average of π_u coincides with the single-point PDF of $u(x,t)$. This suggests a two-step procedure for derivation of PDF equations. First, one derives an equation for $\pi_u(U-u)$. Second, one ensemble-averages (homogenizes) the resulting equation.

The first step relies on rules of differential calculus applied, in the sense of distributions, to the function $\pi_u(U-u)$,

$$\frac{\partial \pi_u}{\partial u} = -\frac{\partial \pi_u}{\partial U}, \qquad \frac{\partial \pi_u}{\partial t} = \frac{\partial \pi_u}{\partial u}\frac{\partial u}{\partial t} = -\frac{\partial \pi_u}{\partial U}\frac{\partial u}{\partial t}, \qquad \frac{\partial \pi_u}{\partial x} = -\frac{\partial \pi_u}{\partial U}\frac{\partial u}{\partial x}. \tag{A.3}$$

Multiplying both sides of (15) with $\partial_U \pi_u$, using the above relations and the sifting property of the delta function, $g(u)\delta(U-u) = g(U)\delta(U-u)$ for any "good" function $g(u)$, we obtain a linear stochastic PDE for π_u,

$$\frac{\partial \pi_u}{\partial t} + k\frac{\partial \pi_u}{\partial x} + r\frac{\partial g(U)\pi_u}{\partial U} = 0. \tag{A.4}$$

For deterministic parameters k and r, the ensemble average of this PDE yields (16), a deterministic equation for the PDF $f_u(U;x,t)$. If one or two of these parameters are random (e.g., k), then ensemble averaging of this PDF is facilitated by a Reynolds decomposition that represents all the independent and dependent variables involved as the sums of their

ensemble means and zero-mean fluctuations about these means, e.g., $k = \langle k \rangle + k'$ and $\pi_u = f_u + \pi'_u$ with $\mathbb{E}[k'] = 0$ and $\mathbb{E}[u'(x,t)] = 0$. For the deterministic r, the ensemble average of (A.4) yields an unclosed PDE for $f_u(U;x,t)$,

$$\frac{\partial f_u}{\partial t} + \langle k \rangle \frac{\partial f_u}{\partial x} + r\frac{\partial g(U) f_u}{\partial U} + \mathcal{C}(f_u) = 0, \qquad \mathcal{C}(f_u) \equiv \mathbb{E}\left[k' \frac{\partial \pi'_u}{\partial x}\right] = \frac{\partial \mathbb{E}[k'\pi'_u]}{\partial x}; \tag{A.5}$$

which is the same as (18).

Appendix B. Derivation of the joint PDF equation

Consider a generalized function

$$\pi_{uk}(U-u, K-k) = \delta(U-u(x,t))\delta(K-k). \tag{B.1}$$

Let $f_{uk}(U,K;x,t)$ denote a joint PDF of the random input k and the random output u at any space-time point (x,t). In analogy to (A.2), $\mathbb{E}[\pi_{uk}] = f_{uk}(U,K;x,t)$. A procedure similar to that used to derive a stochastic PDE (A.4) now yields a deterministic PDE for π_{uk},

$$\frac{\partial \pi_{uk}}{\partial t} + K\frac{\partial \pi_{uk}}{\partial x} + \frac{\partial g(U)\pi_{uk}}{\partial U} = 0. \tag{B.2}$$

The randomness of π_{uk} stems from the random initial state u_0, rather than the model coefficients. Consequently, the averaging of this equation is trivial and exact, and given by (20). This equation is subject to the initial condition $f_{uk}(U,K;x,0) = f_{u_0,k}(U,K;x)$. If $u_0(x)$ and k are mutually independent, then $f_{uk}(U,K;x,0) = f_{u_0}(U;x) f_k(K)$.

The solution can be obtained numerically with a linear solver or, in some cases, analytically. For example, if $g(U) \equiv 0$, then the method of characteristics yields an exact PDF solution

$$f_{uk}(U,K;x,t) = f_{u_0}(U;x-Kt) f_k(K). \tag{B.3}$$

Appendix C. Derivation of closure approximations

One way to approximate the mixed ensemble moment $C_{k\pi} = \langle k'\pi'_u \rangle$ in (A.5) is to subtract (A.5) from (A.4), giving an equation for the random fluctuations $\pi'_u(U;x,t)$,

$$\frac{\partial \pi'_u}{\partial t} + k'\frac{\partial \pi'_u}{\partial x} + k'\frac{\partial f_u}{\partial x} + \langle k \rangle \frac{\partial \pi'_u}{\partial x} + r\frac{\partial g(U)\pi'_u}{\partial U} - \mathcal{C}(f_u) = 0. \tag{C.1}$$

Multiplying (C.1) by k' and taking the ensemble average, we obtain an unclosed PDE for $C_{k\pi}$,

$$\frac{\partial C_{k\pi}}{\partial t} + \frac{\partial \langle k'k'\pi'_u \rangle}{\partial x} + \langle k'k' \rangle \frac{\partial f_u}{\partial x} + \langle k \rangle \frac{\partial C_{k\pi}}{\partial x} - \frac{\partial \langle k'k'\pi'_u \rangle}{\partial x} + \frac{\partial g(U) C_{k\pi}}{\partial U} = 0. \tag{C.2}$$

This equation is closed by neglecting the third-order term, $\langle k'k'\pi'_u \rangle$, which yields a two-dimensional advection equation for $C_{k\pi}$,

$$\frac{\partial C_{k\pi}}{\partial t} + \langle k \rangle \frac{\partial C_{k\pi}}{\partial x} + \frac{\partial g(U) C_{k\pi}}{\partial U} = -\sigma_k^2 \frac{\partial f_u}{\partial x}. \tag{C.3}$$

A closed system of PDEs (A.5) and (C.3) defines the joint dynamics of two dependent variables, $f_u(U;x,t)$ and $C_{k\pi}(U;x,t)$. To reduce this system to a single equation for $f_u(U;x,t)$, we write the solution of (C.3) as

$$C_{k\pi}(U;x,t) = -\sigma_k^2 \int_0^t \int_{-\infty}^{+\infty} \int_{D_u} G(U,V;x,y;t-\tau) \frac{\partial f_u(V;y,\tau)}{\partial y} \mathrm{d}V \mathrm{d}y \mathrm{d}\tau, \tag{C.4}$$

where $G(U,V;x,y;t-\tau)$ is the Green's function for (C.3), defined as the solution of $\partial_t G - \mathbf{U}\cdot\nabla_{\tilde{\mathbf{x}}} G = \delta(U-V)\delta(x-y)\delta(t-\tau)$ subject to the homogeneous initial and boundary conditions. Here, $\tilde{\mathbf{x}} = (x,U)^\top$ and $\mathbf{U} = (\langle k \rangle, g(U))^\top$. Taking the derivative of this solution with respect to x leads directly to (19).

Appendix D. Hyper-parameter tuning

Hyper-parameter tuning can significantly increase the optimization dimensionality. To avoid this, we investigate the dependence of RMSE on two of these parameters: the polynomial order used to represent the coefficients $\boldsymbol{\beta}$ and the KDE bandwidth used to process Monte Carlo realizations. The polynomial order increases the representation accuracy but decreases the sparsity of the discovered PDEs. Fig. D.10 reveals that the sparsity reaches a plateau at polynomial order of 2. This is the order we use in the numerical experiments reported in section 3.

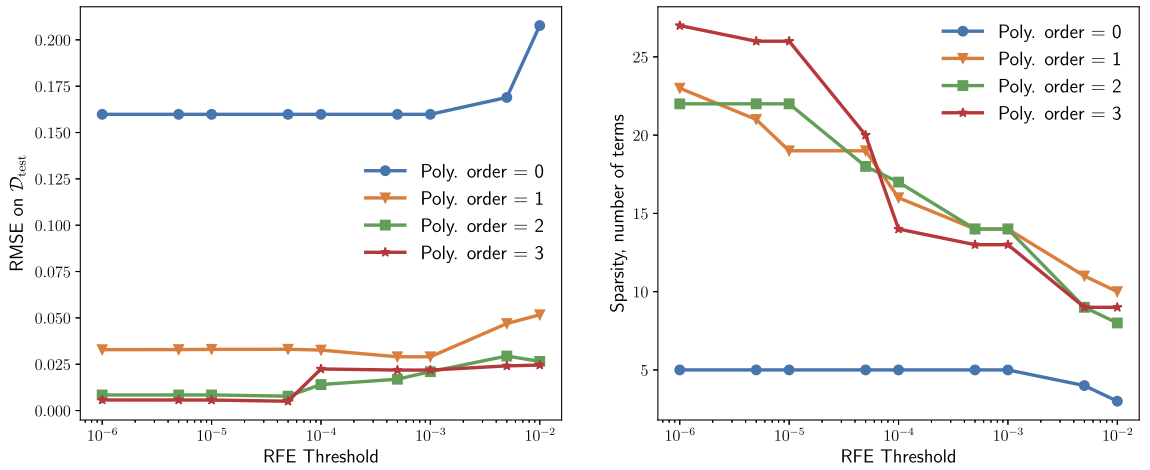

Fig. D.10. RMSE and sparsity in the closure as function of the polynomial order in the advection-reaction closure problem of section 3.2. The results show marginal gains in accuracy and sparsity when the polynomial order is ≥ 2.

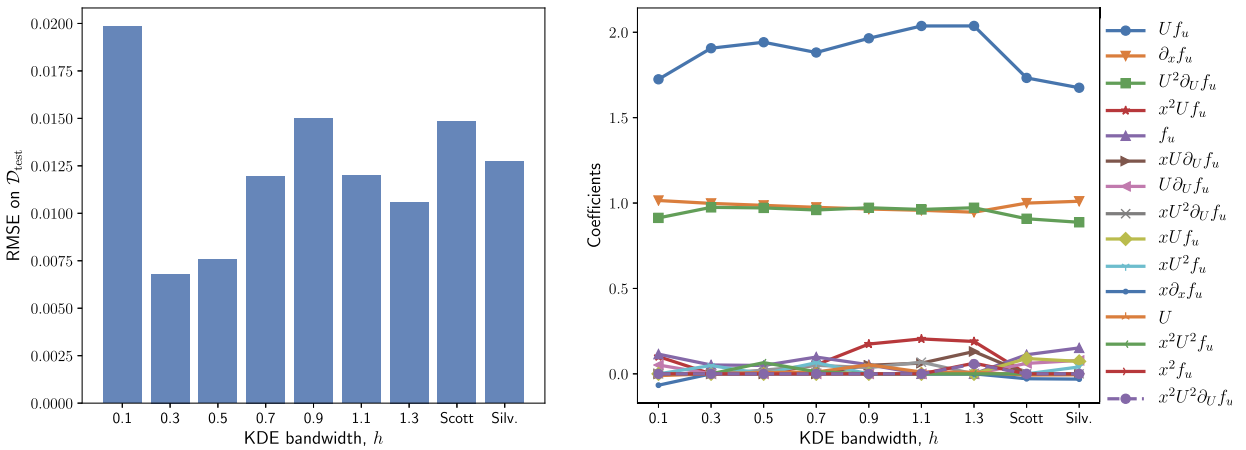

Fig. D.11. Error and coefficients as function of the KDE bandwidth.

Fig. D.11 demonstrates that the KDE bandwidth plays a big role in the accuracy of an estimated PDF f_u and its corresponding derivatives. We found that Scott's or Silverman's rule for estimating the bandwidth does not guarantee optimal coefficients, in comparison to fixed bandwidths. On the other hand, a bandwidth of $h = 1.3$ leads to accurate estimates of the relevant model coefficients, but also adds irrelevant terms with large coefficients.

References

[1] A. Alawadhi, F. Boso, D.M. Tartakovsky, Method of distributions for water-hammer equations with uncertain parameters, Water Resour. Res. 54 (11) (2018) 9398–9411.
[2] D. Barajas-Solano, A.M. Tartakovsky, Probabilistic density function method for nonlinear dynamical systems driven by colored noise, Phys. Rev. E 93 (2016) 052121.
[3] F. Boso, D.M. Tartakovsky, The method of distributions for dispersive transport in porous media with uncertain hydraulic properties, Water Resour. Res. 52 (6) (2016) 4700–4712, https://doi.org/10.1002/2016WR018745.
[4] F. Boso, D.M. Tartakovsky, Information-theoretic approach to bidirectional scaling, Water Resour. Res. 54 (7) (2018) 4916–4928.
[5] F. Boso, D.M. Tartakovsky, Learning on dynamic statistical manifolds, Proc. R. Soc. A 476 (2239) (2020) 20200213, https://doi.org/10.1098/rspa.2020-0213.
[6] F. Boso, D.M. Tartakovsky, Data-informed method of distributions for hyperbolic conservation laws, SIAM J. Sci. Comput. 42 (1) (2020) A559–A583, https://doi.org/10.1137/19M1260773.
[7] F. Boso, S.V. Broyda, D.M. Tartakovsky, Cumulative distribution function solutions of advection-reaction equations with uncertain parameters, Proc. R. Soc. A 470 (2166) (2014) 20140189, https://doi.org/10.1098/rspa.2014.0189.
[8] S.L. Brunton, J.L. Proctor, J.N. Kutz, W. Bialek, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. USA 113 (15) (2016) 3932–3937.
[9] L. Felsberger, P. Koutsourelakis, Physics-constrained, data-driven discovery of coarse-grained dynamics, Commun. Comput. Phys. 25 (2018), https://doi.org/10.4208/cicp.OA-2018-0174.

[10] N. Geneva, N. Zabaras, Quantifying model form uncertainty in Reynolds-averaged turbulence models with Bayesian deep neural networks, J. Comput. Phys. 383 (2019) 125–147.
[11] N. Geneva, N. Zabaras, Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks, J. Comput. Phys. 403 (2020) 109056, https://doi.org/10.1016/j.jcp.2019.109056.
[12] H. Lu, D.M. Tartakovsky, Lagrangian dynamic mode decomposition for construction of reduced-order models of advection-dominated phenomena, J. Comput. Phys. 407 (2020) 109229, https://doi.org/10.1016/j.jcp.2020.109229.
[13] H. Lu, D.M. Tartakovsky, Prediction accuracy of dynamic mode decomposition, SIAM J. Sci. Comput. 42 (3) (2020) A1639–A1662, https://doi.org/10.1137/19M1259948.
[14] T. Maltba, P. Gremaud, D.M. Tartakovsky, Nonlocal PDF methods for Langevin equations with colored noise, J. Comput. Phys. 367 (2018) 87–101.
[15] S.P. Neuman, D.M. Tartakovsky, Perspective on theories of anomalous transport in heterogeneous media, Adv. Water Resour. 32 (5) (2009) 670–680, https://doi.org/10.1016/j.advwatres.2008.08.005.
[16] G. Pang, L. Lu, G.E. Karniadakis, fpinns: Fractional physics-informed neural networks, SIAM J. Sci. Comput. 41 (4) (2019) A2603–A2626.
[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[18] M. Raissi, P. Perdikaris, G.E. Karniadakis, Machine learning of linear differential equations using Gaussian processes, J. Comput. Phys. 348 (2017) 683–693.
[19] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[20] H. Risken, The Fokker-Planck Equation: Methods of Solution and Applications, 2nd edition, Springer-Verlag, New York, 1989.
[21] S. Rudy, A. Alla, S.L. Brunton, J.N. Kutz, Data-driven identification of parametric partial differential equations, SIAM J. Appl. Dyn. Syst. 18 (2) (2019) 643–660.
[22] S.H. Rudy, S.L. Brunton, J.L. Proctor, J.N. Kutz, Data-driven discovery of partial differential equations, Sci. Adv. 3 (4) (2017) 1–7.
[23] H. Schaeffer, Learning partial differential equations via data discovery and sparse optimization, Proc. R. Soc. A 473 (2197) (2017), https://doi.org/10.1098/rspa.2016.0446.
[24] M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data, Science 324 (5923) (2009) 81–85.
[25] M. Schöberl, N. Zabaras, P.-S. Koutsourelakis, Predictive coarse-graining, J. Comput. Phys. 333 (2017) 49–77.
[26] D.W. Scott, On optimal and data-based histograms, Biometrika 66 (3) (1979) 605–610.
[27] J. Sirignano, K. Spiliopoulos, DGM: a deep learning algorithm for solving partial differential equations, J. Comput. Phys. 375 (2018) 1339–1364.
[28] A.M. Tartakovsky, C. Ortiz Marrero, P. Perdikaris, G.D. Tartakovsky, D. Barajas-Solano, Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems, Water Resour. Res. 56 (5) (2020) e2019WR026731.
[29] D.M. Tartakovsky, P.A. Gremaud, Method of distributions for uncertainty quantification, in: R. Ghanem, D. Higdon, H. Owhadi (Eds.), Handbook of Uncertainty Quantification, Springer International Publishing, Switzerland, 2016.
[30] S. Taverniers, D.M. Tartakovsky, Estimation of distributions via multilevel Monte Carlo with stratified sampling, J. Comput. Phys. 419 (2020) 109572, https://doi.org/10.1016/j.jcp.2020.109572.
[31] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. B 58 (1) (1996) 267–288.
[32] D. Venturi, The numerical approximation of nonlinear functionals and functional differential equations, Phys. Rep. 732 (2018) 1–102.
[33] D. Venturi, D.M. Tartakovsky, A.M. Tartakovsky, G.E. Karniadakis, Exact PDF equations and closure approximations for advective-reactive transport, J. Comput. Phys. 243 (2013) 323–343.
[34] P. Wang, A.M. Tartakovsky, D.M. Tartakovsky, Probability density function method for Langevin equations with colored noise, Phys. Rev. Lett. 110 (14) (2013) 140602, https://doi.org/10.1103/PhysRevLett.110.140602.
[35] P. Wang, D.M. Tartakovsky, J.K.D. Jarman, A.M. Tartakovsky, CDF solutions of Buckley–Leverett equation with uncertain parameters, Multiscale Model. Simul. 11 (1) (2013) 118–133, https://doi.org/10.1137/120865574.
[36] C.L. Winter, D.M. Tartakovsky, A. Guadagnini, Moment equations for flow in highly heterogeneous porous media, Surv. Geophys. 24 (1) (2003) 81–106.
[37] K. Wu, D. Xiu, Data-driven deep learning of partial differential equations in modal space, J. Comput. Phys. 408 (2020) 109307, https://doi.org/10.1016/j.jcp.2020.109307.
[38] H.-J. Yang, F. Boso, H.A. Tchelepi, D.M. Tartakovsky, Probabilistic forecast of single-phase flow in porous media with uncertain properties, Water Resour. Res. 55 (11) (2019) 8631–8645.
[39] M. Ye, S.P. Neuman, A. Guadagnini, D.M. Tartakovsky, Nonlocal and localized analyses of conditional mean transient flow in bounded, randomly heterogeneous porous media, Water Resour. Res. 40 (2004) W05104, https://doi.org/10.1029/2003WR002099.