Chaos
An Interdisciplinary Journal of Nonlinear Science

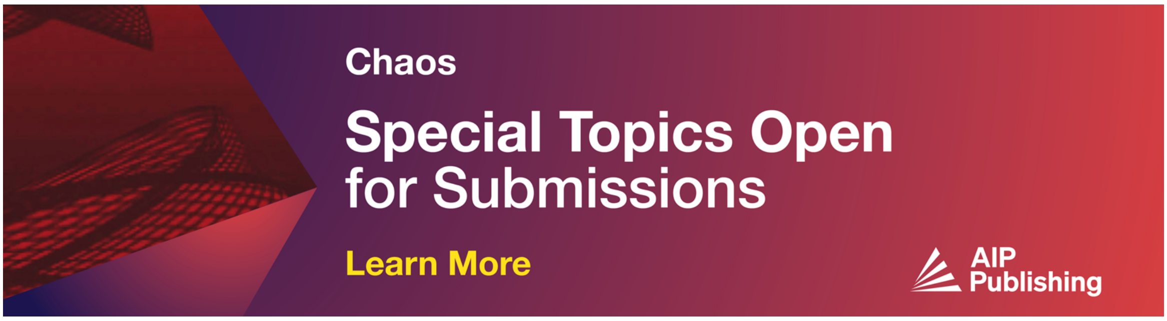
RESEARCH ARTICLE | JUNE 24 2022

Discovery of interpretable structural model errors by combining Bayesian sparse regression and data assimilation: A chaotic Kuramoto–Sivashinsky test case FREE

Special Collection: Theory-informed and Data-driven Approaches to Advance Climate Sciences

Rambod Mojgani ✉ ; Ashesh Chattopadhyay ; Pedram Hassanzadeh

Chaos 32, 061105 (2022)
https://doi.org/10.1063/5.0091282



Discovery of interpretable structural model errors by combining Bayesian sparse regression and data assimilation: A chaotic Kuramoto–Sivashinsky test case

Cite as: Chaos **32**, 061105 (2022); doi: 10.1063/5.0091282
Submitted: 13 March 2022 · Accepted: 1 June 2022 ·
Published Online: 24 June 2022

Rambod Mojgani,[a] Ashesh Chattopadhyay, and Pedram Hassanzadeh

AFFILIATIONS

Department of Mechanical Engineering, Rice University, Houston, Texas 77005, USA

Note: This article is part of the Focus Issue, Theory-informed and Data-driven Approaches to Advance Climate Sciences.
[a]**Author to whom correspondence should be addressed:** rm99@rice.edu

ABSTRACT

Models of many engineering and natural systems are imperfect. The discrepancy between the mathematical representations of a true physical system and its imperfect model is called the model error. These model errors can lead to substantial differences between the numerical solutions of the model and the state of the system, particularly in those involving nonlinear, multi-scale phenomena. Thus, there is increasing interest in reducing model errors, particularly by leveraging the rapidly growing observational data to understand their physics and sources. Here, we introduce a framework named MEDIDA: Model Error Discovery with Interpretability and Data Assimilation. MEDIDA only requires a working numerical solver of the model and a small number of noise-free or noisy sporadic observations of the system. In MEDIDA, first, the model error is estimated from differences between the observed states and model-predicted states (the latter are obtained from a number of one-time-step numerical integrations from the previous observed states). If observations are noisy, a data assimilation technique, such as the ensemble Kalman filter, is employed to provide the analysis state of the system, which is then used to estimate the model error. Finally, an equation-discovery technique, here the relevance vector machine, a sparsity-promoting Bayesian method, is used to identify an interpretable, parsimonious, and closed-form representation of the model error. Using the chaotic Kuramoto–Sivashinsky system as the test case, we demonstrate the excellent performance of MEDIDA in discovering different types of structural/parametric model errors, representing different types of missing physics, using noise-free and noisy observations.

The discovery of the governing equations of physical systems is often based on the first principles, which has been the origin of most advances in science and engineering. However, in many important applications, some of the underlying physical processes are not well understood and, therefore, are missing or poorly represented in the mathematical models of such systems. Accordingly, these *imperfect models* ("models" hereafter) can merely closely track the dynamics of the *true physical system* ("system" hereafter) while failing to exactly represent it. Global climate models (GCMs) and numerical weather prediction (NWP) models are prominent examples of such models, often suffering from parametric and structural errors. The framework proposed here integrates machine learning (ML) and data assimilation (DA) techniques to discover close-form, interpretable representations of these model errors. This framework can leverage the ever-growing abundance of observational data to reduce the errors in models of nonlinear dynamical systems, such as the climate system.

I. INTRODUCTION

The difference between the solution of the model and the system becomes prominent in many problems involving complex, nonlinear, multi-scale phenomena such as those in engineering,[1–3] thermo-fluids,[4,5] and climate/weather prediction;[6,7] see Levine and Stuart[8] for an insightful overview. The deviation of the model from the system is called, in different communities, model uncertainty (structural and/or parametric), model discrepancy, model inadequacy, missing dynamics, or "model error"; hereafter, the latter is used.

Recently, many studies have focused on leveraging the rapid advances in ML techniques and availability of data (e.g., high-quality observations) to develop more accurate models. Several main approaches include learning fully data-driven (equation-free) models[9–15] or data-driven subgrid-scale closures.[16–24] In a third approach, corrections to the state or its temporal derivative (tendency) are learned from the deviation of the model predictions from the observations.[25–29] More specifically, the model is initialized with the observed state, integrated forward in time, and the difference between the predicted state and the observation at the later time is computed. Repeated many times, a correction scheme, e.g., a deep neural network (DNN), can be trained to nudge the model's predicted trajectory (or tendency) to that of the system. To deal with observations with noise, a number of studies have integrated DA with DNNs.[30–36]

These studies often used DNNs, showing promising results for a variety of test cases. However, while powerfully expressive, DNNs are currently hard to interpret and often fail to generalize (especially extrapolate) when the systems' parameters change.[17,24,37,38] The interpretability of models is crucial for robust, reliable, and generalizable decision-critical predictions or designs.[3,7] Posing the task of system identification as a linear regression problem, based on a library of nonlinear terms, exchanges the expressivity of DNNs for the sake of interpretability.[39,40] The closed-form representation of the identified models and their parsimony (i.e., sparsity in the space of the employed library) is the key advantage of these methods, leading to highly interpretable models. A number of studies have used such methods to discover closed-form full models or closures.[40–46] While the results are promising, noisy data, especially in the chaotic regimes, significantly degrades the quality of the discovered models.[40,43,45,46]

So far, there has not been any work on discovering closed-form representation of model error using the differences between model predictions and observations (approach 3) or on combining the sparsity-promoting regression methods with DA to alleviate observation noise. Here, we introduce MEDIDA (Model Error Discovery with Interpretability and Data Assimilation), a general-purpose, data-efficient, non-intrusive framework to discover the structural (and parametric) model errors in the form of missing/incorrect terms of partial differential equations (PDEs). MEDIDA uses differences between predictions from a working numerical solver of the model and noisy sporadic (sparse in time) observations. The framework is built on two main ideas:

(i) discovering interpretable and closed-form model errors using relevance vector machine (RVM), a sparsity-promoting Bayesian regression method,[47] and
(ii) reducing the effects of observation noise using DA methods such as ensemble Kalman filter (EnKF) to generate the "analysis state."

Below, we present the problem statement and introduce MEDIDA. Subsequently, its performance is demonstrated using a chaotic Kuramoto–Sivashinsky (KS) test case.

II. PROBLEM STATEMENT

Suppose the exact mathematical representation of a system is a nonlinear PDE,

$$\partial_t u(t) = g(u(t)), \tag{1}$$

in a continuous domain. Here, $u(t)$ is the state variable at time t. While (1) is unknown, we assume to have access to sporadic pairs of observations of the state variable u^o. These observations might be contaminated by measurement noise. The set of observed states at t_i is denoted as $\{u^o(t_i - \Delta t_i), u^o(t_i)\}_{i=1}^n$. Note that t_i do not have to be equally spaced. Furthermore, Δt_i should be similar for all i but do not have to be the same (hereafter, we use $\forall i, \Delta t_i = \Delta t$ for convenience).

Moreover, suppose that we have a *model* of the system,

$$\partial_t u(t) = f(u(t)). \tag{2}$$

Without loss of generality,[8] we assume that the deviation of (2) from (1) is additive; we further assume that the deviation is only a function of the state.[30,31] Therefore, the model error is

$$h(u(t)) := g(u(t)) - f(u(t)). \tag{3}$$

Our goal is to find a closed-form representation of h given a working numerical solver of (2) and noisy or noise-free observations $\{u^o(t_i - \Delta t), u^o(t_i)\}_{i=1}^n$.

III. FRAMEWORK: MEDIDA

MEDIDA has three main steps (Fig. 1): Step (1) Collecting sporadic observations of the system and numerically integrating the model forward for short-time (Sec. III A); Step (2) Construction and solving a regression problem using RVM,[47] which leads to the sparse identification of the model error (Sec. III B); and Step (3) If observations are noisy, following step 1, stochastic EnKF[48,49] is used to estimate an analysis state, an estimate of the system's state, for Step 2 (Sec. III C). We emphasize that at no point does MEDIDA need any knowledge of the system (1). Also, note that other equation-discovery techniques and ensemble-based DA methods can be used in steps 2 and 3.

A. Interpretable model error

Consider the discretized (2),

$$\boldsymbol{u}^m(t_i) = \boldsymbol{f}(\boldsymbol{u}^o(t_i - \Delta t)). \tag{4}$$

For brevity, we use the notation of an explicit scheme but the scheme can also be implicit, as shown in the test case. The domain is discretized on the grid of $\boldsymbol{x} \in \mathbb{R}^M$. The observation at $t_i - \Delta t$, i.e., $u^o(t_i - \Delta t)$, is the initial condition and $\boldsymbol{u}^m(t_i)$ is the predicted state at time t_i [Fig. 1(a)].

At each time t_i, subtracting the state predicted by the model ($\boldsymbol{u}^m(t_i)$) from the observed state ($\boldsymbol{u}^o(t_i)$) results in an approximation of the model error at t_i [Fig. 1(b)],

$$\boldsymbol{h}(t_i) \approx \Delta \boldsymbol{u}_i := (\boldsymbol{u}^o(t_i) - \boldsymbol{u}^m(t_i))/\Delta t. \tag{5}$$

The difference between the states of the system and the model is similar to the analysis increment in the DA literature, where the best estimate of the state replaces $\boldsymbol{u}^o(t_i)$ in (5) (see Sec. III C). The idea

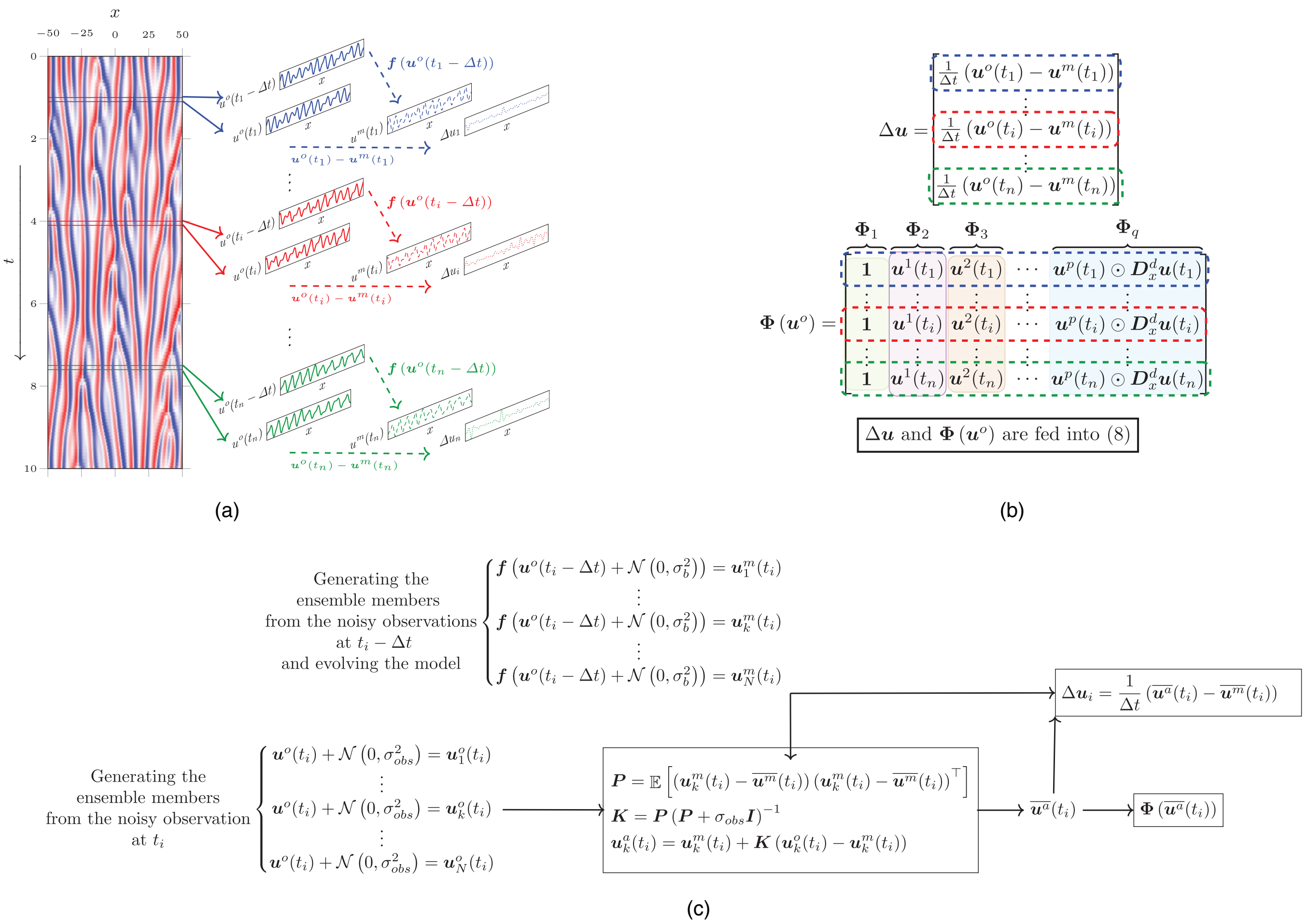

x
−50 −25 0 25 50
t
0
2
4
6
8
10
$u^o(t_1-\Delta t)$
$u^o(t_1)$
$\boldsymbol{f}(\boldsymbol{u}^o(t_1-\Delta t))$
$u^m(t_1)$
Δu_1
$\boldsymbol{u}^o(t_1)-\boldsymbol{u}^m(t_1)$
$\boldsymbol{f}(\boldsymbol{u}^o(t_i-\Delta t))$
Δu_i
$\boldsymbol{u}^o(t_i)-\boldsymbol{u}^m(t_i)$
$\boldsymbol{f}(\boldsymbol{u}^o(t_n-\Delta t))$
Δu_n
$\boldsymbol{u}^o(t_n)-\boldsymbol{u}^m(t_n)$
(a)
$\Delta\boldsymbol{u}=\frac{1}{\Delta t}(\boldsymbol{u}^o(t_1)-\boldsymbol{u}^m(t_1)) \;\vdots\; \frac{1}{\Delta t}(\boldsymbol{u}^o(t_i)-\boldsymbol{u}^m(t_i)) \;\vdots\; \frac{1}{\Delta t}(\boldsymbol{u}^o(t_n)-\boldsymbol{u}^m(t_n))$
$\boldsymbol{\Phi}_1$ $\boldsymbol{\Phi}_2$ $\boldsymbol{\Phi}_3$ $\boldsymbol{\Phi}_q$
$\boldsymbol{\Phi}(\boldsymbol{u}^o)=$ $\boldsymbol{1}$ $\boldsymbol{u}^1(t_1)$ $\boldsymbol{u}^2(t_1)$ $\cdots$ $\boldsymbol{u}^p(t_1)\odot\boldsymbol{D}_x^d\boldsymbol{u}(t_1)$
$\boldsymbol{1}$ $\boldsymbol{u}^1(t_i)$ $\boldsymbol{u}^2(t_i)$ $\cdots$ $\boldsymbol{u}^p(t_i)\odot\boldsymbol{D}_x^d\boldsymbol{u}(t_i)$
$\boldsymbol{1}$ $\boldsymbol{u}^1(t_n)$ $\boldsymbol{u}^2(t_n)$ $\cdots$ $\boldsymbol{u}^p(t_n)\odot\boldsymbol{D}_x^d\boldsymbol{u}(t_n)$
$\Delta\boldsymbol{u}$ and $\boldsymbol{\Phi}(\boldsymbol{u}^o)$ are fed into (8)
(b)
Generating the ensemble members from the noisy observations at $t_i-\Delta t$ and evolving the model
$\boldsymbol{f}(\boldsymbol{u}^o(t_i-\Delta t)+\mathcal{N}(0,\sigma_b^2))=\boldsymbol{u}_1^m(t_i)$
$\boldsymbol{f}(\boldsymbol{u}^o(t_i-\Delta t)+\mathcal{N}(0,\sigma_b^2))=\boldsymbol{u}_k^m(t_i)$
$\boldsymbol{f}(\boldsymbol{u}^o(t_i-\Delta t)+\mathcal{N}(0,\sigma_b^2))=\boldsymbol{u}_N^m(t_i)$
Generating the ensemble members from the noisy observation at t_i
$\boldsymbol{u}^o(t_i)+\mathcal{N}(0,\sigma_{obs}^2)=\boldsymbol{u}_1^o(t_i)$
$\boldsymbol{u}^o(t_i)+\mathcal{N}(0,\sigma_{obs}^2)=\boldsymbol{u}_k^o(t_i)$
$\boldsymbol{u}^o(t_i)+\mathcal{N}(0,\sigma_{obs}^2)=\boldsymbol{u}_N^o(t_i)$
$\boldsymbol{P}=\mathbb{E}\left[(\boldsymbol{u}_k^m(t_i)-\overline{\boldsymbol{u}^m}(t_i))(\boldsymbol{u}_k^m(t_i)-\overline{\boldsymbol{u}^m}(t_i))^\top\right]$
$\boldsymbol{K}=\boldsymbol{P}(\boldsymbol{P}+\sigma_{obs}\boldsymbol{I})^{-1}$
$\boldsymbol{u}_k^a(t_i)=\boldsymbol{u}_k^m(t_i)+\boldsymbol{K}(\boldsymbol{u}_k^o(t_i)-\boldsymbol{u}_k^m(t_i))$
$\Delta\boldsymbol{u}_i=\frac{1}{\Delta t}(\overline{\boldsymbol{u}^a}(t_i)-\overline{\boldsymbol{u}^m}(t_i))$
$\overline{\boldsymbol{u}^a}(t_i)$
$\boldsymbol{\Phi}(\overline{\boldsymbol{u}^a}(t_i))$
(c)

FIG. 1. The schematic of MEDIDA presented in the context of the test case in Sec. IV. (a) Step 1: n pairs of state variables are sampled uniformly or non-uniformly at t_i from observations [of the system (1)] to obtain $\{\boldsymbol{u}^o(t_i-\Delta t), \boldsymbol{u}^o(t_i)\}_{i=1}^n$. Model (4) is integrated numerically for one time step from each $\boldsymbol{u}^o(t_i-\Delta t)$ to predict $\boldsymbol{u}^m(t_i)$. For each sample, at t_i, the difference between the predicted and observed state is used to compute $\Delta\boldsymbol{u}_i$ (5), an approximation of model error. (b) Step 2: $\{\Delta\boldsymbol{u}_i\}_{i=1}^n$ are stacked to form $\Delta\boldsymbol{u}$. Moreover, the library of bases $\Phi(\boldsymbol{u}^o)$, consisting of selected q linear and nonlinear bases evaluated at $\{\boldsymbol{u}^o(t_i)\}_{i=1}^n$, is formed. Here, the library in (7) is used but the bases with any arbitrary functional form could be used depending on the anticipated form of the model error. $\Delta\boldsymbol{u}$ and Φ are then fed into the RVM (or other equation-discovery methods) to compute $\boldsymbol{c}^*$ for minimizing the loss function (8). The corrected model is then identified using $\boldsymbol{c}^*$ as Eq. (9). (c) Step 3: If observations are noisy, DA is used to provide an "analysis state" $\{\boldsymbol{u}^a(t_i)\}_{i=1}^n$ to be used in place of $\{\boldsymbol{u}^o(t_i)\}_{i=1}^n$ for computing $\Delta\boldsymbol{u}$ and Φ in Steps 1 and 2. Here, we use a stochastic EnKF for DA: for each sample i, the noisy observation is perturbed by Gaussian white noise with inflated standard deviation σ_b to generate an ensemble of size N. Each ensemble member k is numerically evolved by the model, $\left\{\boldsymbol{f}\left(\boldsymbol{u}_k^o(t_i-\Delta t)\right)\right\}_{k=1}^N$, to generate the ensemble of model prediction $\left\{\boldsymbol{u}_k^m(t_i)\right\}_{k=1}^N$. Observations at time t_i are also perturbed by Gaussian white noise with standard deviation σ_{obs} to generate the ensemble $\left\{\boldsymbol{u}_k^o(t_i)\right\}_{k=1}^N$. These two ensembles are then used to compute the background covariance matrix $\boldsymbol{P}$, Kalman gain $\boldsymbol{K}$, and, finally, the analysis state $\overline{\boldsymbol{u}^a}(t_i)$; overbars indicate ensemble mean and $\mathbb{E}[.]$ denotes expectation. $\Delta\boldsymbol{u}$ and Φ in steps 1 and 2 are then computed using $\overline{\boldsymbol{u}^a}(t_i)$.

of accounting for model error through the analysis increment was first introduced by Leith[50] and is at the core of many recent applications of ML aiming to *nudge* the model to its correct trajectory[31] or to account for the model error.[51,52] However, in this manuscript, we aim to *discover* an interpretable representation of model error, $h(u(t))$.

Note that to accurately discover $h(u(t))$ as defined in (3), $\Delta\boldsymbol{u}_i$ should be *dominated by the model error*, and the contributions from numerical errors [in obtaining $\boldsymbol{u}^m(t_i)$] and measurement errors [in $\boldsymbol{u}^o(t_i)$] should be minimized as much as possible. As discussed in Sec. III C, DA can be utilized to reduce the contributions from the observation noise. Computing $\Delta\boldsymbol{u}_i$ after only one Δt prevents the accumulation of numerical error from integrating (4). Moreover, this avoids nonlinear growth of the model error and its further entanglement with truncation errors, which can complicate the discovery of just model error. Therefore, in MEDIDA, the model time step and observation intervals are set to be the same. Note that while the size of Δt could be restricted by the availability of the observation pairs, increasing M can be used to reduce truncation errors from spatial discretizations.

Integrating (4) from $\boldsymbol{u}^o(t_i - \Delta t)$ and computing $\Delta \boldsymbol{u}_i$ from (5) are repeated for n samples (step 1), and the vectors of model error are stacked to form $\Delta \boldsymbol{u} \in \mathbb{R}^{nM}$ (step 2) [see Figs. 1(a) and 1(b)]. Similar to past studies,[40,41,43,53] we further assume that the model error spans over the space of the bases or training vectors, i.e.,

$$h(u(t)) = c_1\phi_1 + \cdots + c_q\phi_q, \tag{6}$$

where ϕ_i is a linear or nonlinear function of the state variable, the building block of the library of training bases $\{\phi_k(u(t))\}_{k=1}^{q}$. The selection of the bases should be guided by the physical understanding of the system and the model.

Here, in the discretized form, we assume that $\boldsymbol{\Phi}(\boldsymbol{u})$ is a linear combination of polynomial terms up to pth-order and spatial derivatives up to dth-order, i.e.,

$$\boldsymbol{\Phi}_k(\boldsymbol{u}) = \boldsymbol{u}^r \odot \boldsymbol{D}_x^s \boldsymbol{u}; \quad r \in \{0,\ldots,p\},\ s \in \{0,\ldots,d\}, \tag{7}$$

where in $\boldsymbol{u}^r$, each element of $\boldsymbol{u}$ is raised to the power of r, $\boldsymbol{D}_x^s\boldsymbol{u}$ denotes the sth spatial derivative of $\boldsymbol{u}$, and $\odot$ is the element-wise (Hadamard) product. Therefore, the model error lies on the space of library of the training bases evaluated using the observed state at each t_i, $\boldsymbol{\Phi}\left(\boldsymbol{u}^o, t_i\right) \in \mathbb{R}^{M\times q}$. For all the n samples, the library of the bases are stacked to form $\boldsymbol{\Phi}\left(\boldsymbol{u}^o\right) \in \mathbb{R}^{nM\times q}$ [Fig. 1(b)].

At the end of step 2, the regression problem for discovery of model error is formulated as

$$\boldsymbol{c}^* = \underset{\boldsymbol{c}}{\operatorname{argmin}} \left\| \Delta\boldsymbol{u} - \boldsymbol{\Phi}\left(\boldsymbol{u}^o\right)\boldsymbol{c} \right\|_2, \tag{8}$$

where $\boldsymbol{c} \in \mathbb{R}^q$ is the vector of coefficients corresponding to the training bases, and $\boldsymbol{c}^* = \left[c_1^*, c_2^*, \ldots, c_q^*\right]^\top$ is a minimizer of the regression problem, i.e., the vector of coefficients of the model error. Finally, the discovered (corrected) model is identified as

$$\partial_t u(t) = f(u(t)) + \Sigma_{k=1}^{q} c_k^* \phi_k(u(t))\,. \tag{9}$$

B. Solution of the regression problem

In this study, we use RVM[47] to compute $\boldsymbol{c}^*$ in (8) for inputs of $\Delta\boldsymbol{u}$ and $\boldsymbol{\Phi}(\boldsymbol{u}^o)$. RVM leads to a sparse identification of columns of $\boldsymbol{\Phi}$ with posterior distribution of the corresponding weights, i.e., the *relevant vectors*. See Zhang and Lin[41] for a detailed discussion in the context of PDEs discovery.

The breakthrough in equation-discovery originates in the introduction of parsimony.[39] While the original least absolute shrinkage and selection operator (LASSO)-type regularization has yielded promising results in a broad range of applications, RVMs have been found to achieve the desired sparsity by a more straightforward hyper-parameter tuning and a relatively lower sensitivity to noise.[41,44,54] The hyper-parameter in RVMs is a threshold to prune the bases with higher posterior variance, i.e., highly uncertain bases are removed. To avoid over-fitting, we choose the hyper-parameter as a compromise between the sparsity and accuracy of the corrected model at the elbow of the L-curve, following Mangan *et al.*[55]

C. Data assimilation

Steps 1 and 2 described above suffice to find an accurate corrected model (9) if the fully observed state is noise-free. However, noise can cause substantial challenges in the sparse identification of nonlinear systems, a topic of ongoing research.[40,42,43,54,56] In MEDIDA, we propose to use DA, a powerful set of tools to deal with noisy and partial observation (spatiotemporal sparsity). Here, we use stochastic EnKF [57] [Fig. 1(c)]. In this study, we assume that observations of the full state are available; i.e., $\mathscr{H}(u(t)) = u(t) \in \mathbb{R}^M$, where $\mathscr{H}(.)$ is the observation operator. Dealing with partial observations and more general forms of observation errors (e.g., as in Hamilton *et al.*[58]) remains subject of future investigations.

The result of this step is the analysis state used to construct the vector of model error and the library of the bases [Fig. 1(b)].

At each sample time $t_i - \Delta t$, the observations are further perturbed with Gaussian white noise to obtain an ensemble of size N of the initial conditions,

$$\boldsymbol{u}_k^o(t_i - \Delta t) = \boldsymbol{u}^o(t_i - \Delta t) + \mathscr{N}\left(0, {\sigma_b}^2\right), \tag{10}$$

where σ_b is standard deviation of the observation noise (σ_{obs}) times an inflation factor,[48,59] and k denotes the kth ensemble member.

Subsequently, the model is evolved for each of these ensemble members, i.e., $\boldsymbol{u}_k^m(t_i) = \boldsymbol{f}\left(\boldsymbol{u}_k^o(t_i - \Delta t)\right)$. This ensemble of model prediction is used to construct the background covariance matrix as

$$\boldsymbol{P} = \mathbb{E}\left[\left(\boldsymbol{u}_k^m(t_i) - \overline{\boldsymbol{u}^m}(t_i)\right)\left(\boldsymbol{u}_k^m(t_i) - \overline{\boldsymbol{u}^m}(t_i)\right)^\top\right], \tag{11}$$

where $\overline{\boldsymbol{u}^m}(t_i)$ is the ensemble average of the model prediction and $\mathbb{E}[.]$ denotes the expected value. Having the observation operator to be linear, and non-sparse, the Kalman gain, $\boldsymbol{K} \in \mathbb{R}^{M\times M}$, is then calculated as

$$\boldsymbol{K} = \boldsymbol{P}(\boldsymbol{P} + \sigma_{obs}\boldsymbol{I})^{-1}, \tag{12}$$

where $\boldsymbol{I} \in \mathbb{R}^{M\times M}$ is the identity matrix.

For each sample at the prediction time t_i, the kth member of the ensemble of the observations is generated as

$$\boldsymbol{u}_k^o(t_i) = \boldsymbol{u}^o(t_i) + \mathscr{N}\left(0, {\sigma_{obs}}^2\right). \tag{13}$$

Note that while the EnKF can be used for non-Gaussian observation noise, the method would be sub-optimal and may require further modification.[60] Hereon, we limit our experiments to Gaussian noise.

Finally, the kth member of the ensemble of the analysis state is

$$\boldsymbol{u}_k^a(t_i) = \boldsymbol{u}_k^m(t_i) + \boldsymbol{K}\left(\boldsymbol{u}_k^o(t_i) - \boldsymbol{u}_k^m(t_i)\right). \tag{14}$$

The process of EnKF assimilates the noisy observations to the background forecast obtained from the model, $\boldsymbol{u}_k^m(t_i)$. Subsequently, the ensemble average of the analysis states, $\overline{\boldsymbol{u}^a}(t_i)$, is treated as a close estimate of the true state of the system, $\boldsymbol{u}^o(t_i)$ in Sec. III A.

Similarly, the difference between the ensemble averages of the analysis state and the model prediction, i.e.,

$$\Delta\boldsymbol{u}(t_i) = (\overline{\boldsymbol{u}^a}(t_i) - \overline{\boldsymbol{u}^m}(t_i))/\Delta t, \tag{15}$$

replaces (5) and is fed into the algorithm of Sec. III A.

There are a few points about this DA step that needs to be further clarified. First, as discussed in Sec. III A, the model is evolved for only one time step. Although this may hinder the traditional spin-up period used in EnKF, as a compromise, we resort to a large ensemble and inflation to ensure a successful estimation of the state.

Second, for MEDIDA to work well, (15) should be dominated by model error. However, the inevitable presence of other sources of error, notably the observation errors, generally leads to an overestimation of the actual model error (see Carrassi and Vannitsem[51] for discussions and remedies).

Finally, note that ensemble Kalman smoother (EnKS)[61] is shown to be more suitable for offline pre-processing of the data used in ML models;[33] however, in MEDIDA, the model is evolved for a single time step; therefore, we expect that EnKS and EnKF to behave similarly.

D. Performance metric

To quantify the accuracy of MEDIDA, we use the normalized distance between the vector of coefficients, defined as

$$\varepsilon_m = \frac{\|\boldsymbol{c}_s - \boldsymbol{c}_m\|_2}{\|\boldsymbol{c}_s\|_2} \quad \text{and} \quad \varepsilon^* = \frac{\|\boldsymbol{c}_s - \boldsymbol{c}^*\|_2}{\|\boldsymbol{c}_s\|_2}, \tag{16}$$

where $\boldsymbol{c}_s$, $\boldsymbol{c}_m$, $\boldsymbol{c}^*$ are coefficient vectors of system (1), the model (2), and the corrected model (9).[40,43] The ith element of the coefficient vector is a scalar corresponding to the ith term in the equation. Quantitatively, the goal is to achieve $\varepsilon^* \ll \varepsilon_m$. Note that the implicit assumption in this metric is that the model and the model error can be expressed on the space spanned by the bases of the library.

IV. TEST CASE: KS EQUATION

To evaluate the performance of MEDIDA, we use a chaotic KS system, a challenging test case for system identification, particularly from noisy observations.[40,43] The PDE is

$$\partial_t u + u\partial_x u + \partial_x^2 u + \partial_x^4 u = 0, \tag{17}$$

where $u\partial_x u$ is convection, $\partial_x^2 u$ is anti-diffusion (destabilizing), and $\partial_x^4 u$ is hyper-diffusion (stabilizing). The domain is periodic, $u(0,t) = u(L,t)$. We use $L = 32\pi$, which leads to a highly chaotic system.[9,62]

Here, (17) is the system. We have created nine (imperfect) models as shown in Table I. In cases 1–3, one of the convection, anti-diffusion, or hyper-diffusion terms is entirely missing (i.e., structural uncertainty). In cases 4–6, some or all of the coefficients of the system are incorrect (parametric uncertainty). Finally, in cases 7-9, a mixture of parametric and structural uncertainties, with missing and extra terms, is present.

The system and the models are numerically solved using different time-integration schemes and time-step sizes to introduce additional challenges and a more realistic test case. To generate $\boldsymbol{u}^o$, (17) is integrated with the exponential time-differencing fourth-order Runge–Kutta[63] with time-step Δt^o [in Fig. 1(a), $1t = 10^3 \Delta t^o$]. The models are integrated with second-order Crank–Nicolson and Adams–Bashforth schemes with $\Delta t^m = \Delta t = 5\Delta t^o$. For both system and models, $M = 1024$ Fourier modes are used (different discretizations or M for system and models could have been used too). $\boldsymbol{u}^o$ is collected after a spin-up period of $\tau = 200\Delta t$ and uniformly at $t_i = i\tau$. $\boldsymbol{\Phi}$, is constructed with $d = 4$ and $p = 4$ as defined in (7). Open-source codes are used to generate $\boldsymbol{\Phi}$ [40] and solve (8) using RVM.[44]

A. Noise-free observations

First, we examine the performance of MEDIDA for noise-free observations (only steps 1 and 2). The motivation is twofold: (i) to test steps 1 and 2 before adding the complexity of noisy observations/DA, and (ii) in many applications, an estimate of $\boldsymbol{u}^o$ is already available (see Sec. V).

Here, $\boldsymbol{u}^o$ are from the numerical solutions of (17). The models, MEDIDA-corrected models, and the corresponding errors for $n = 100$ samples show $\varepsilon^* < 2\%$, between 40 and 200 times improvement compared to ε_m (Table I).

To conclude, MEDIDA is a data-efficient framework. Its performance is insensitive to n such that ε^* changes by 0.1% for $n \in \{10, 100, 1000\}$. There are two reasons for this: (i) RVM is known to be data-efficient (compared to DNNs)[64] and (ii) each grid point at t_i provides a data-point, i.e., a row in $\Delta\boldsymbol{u}$ and $\boldsymbol{\Phi}$, although these are not all independent/uncorrelated data-points.

B. Noisy observations

Next, we examine the performance of MEDIDA for noisy observations, obtained from adding noise to the numerical solution of KS: $u^o(t) = u(t) + \mathcal{N}\left(0, {\sigma_{obs}}^2\right)$. Here, $\sigma_{obs}/\sigma_u = 0.01$, where

TABLE I. MEDIDA-corrected models from noise-free and noisy observations of the KS system: $0 = \partial_t u + u\partial_x u + \partial_x^2 u + \partial_x^4 u$. ε_m and ε^* are defined in (16). For the noisy cases, $\sigma_b = 20\sigma_{obs}$, and the ensemble size is $N = 10M$, except for cases 3, 6, and 9 with $N = 200M$, $100M$, and $500M$, respectively.

	Model: $0 = \partial_t u +$	ε_m (%)	Corrected model (noise-free): $0 = \partial_t u +$	ε^* (%)	Corrected model (noisy): $0 = \partial_t u +$	ε^* (%)
1	$\partial_x^2 u + \partial_x^4 u$	57.74	$0.97u\partial_x u + \partial_x^2 u + \partial_x^4 u$	1.50	$0.98u\partial_x u + \partial_x^2 u + \partial_x^4 u$	1.43
2	$u\partial_x u + \partial_x^4 u$	57.74	$u\partial_x u + 0.99\partial_x^2 u + \partial_x^4 u$	0.39	$u\partial_x u + 0.98\partial_x^2 u + \partial_x^4 u$	0.98
3	$u\partial_x u + \partial_x^2 u$	57.74	$u\partial_x u + \partial_x^2 u + 1.02\partial_x^4 u$	1.08	$u\partial_x u + \partial_x^2 u + 1.02\partial_x^4 u$	0.98
4	$0.5u\partial_x u + \partial_x^2 u + \partial_x^4 u$	28.87	$0.99u\partial_x u + \partial_x^2 u + \partial_x^4 u$	0.75	$0.99u\partial_x u + \partial_x^2 u + \partial_x^4 u$	0.63
5	$u\partial_x u + 0.5\partial_x^2 u + \partial_x^4 u$	28.87	$u\partial_x u + 1.00\partial_x^2 u + \partial_x^4 u$	0.14	$u\partial_x u + 0.98\partial_x^2 u + \partial_x^4 u$	0.99
6	$u\partial_x u + \partial_x^2 u + 0.5\partial_x^4 u$	28.87	$u\partial_x u + \partial_x^2 u + 1.00\partial_x^4 u$	0.14	$u\partial_x u + \partial_x^2 u + 1.00\partial_x^4 u$	0.15
7	$0.5u\partial_x u + 2\partial_x^4 u$	86.60	$0.97u\partial_x u + 1.00\partial_x^2 u + \partial_x^4 u$	1.55	$0.98u\partial_x u + 0.98\partial_x^2 u + \partial_x^4 u$	1.63
8	$u\partial_x u + \partial_x^2 u + 0.5\partial_x^3 u + \partial_x^4 u$	28.87	$u\partial_x u + \partial_x^2 u + \partial_x^4 u$	0.28	$u\partial_x u + \partial_x^2 u - 0.01\partial_x^3 u + \partial_x^4 u$	0.83
9	$u\partial_x u + \partial_x^2 u + 0.5\partial_x^3 u$	64.55	$u\partial_x u + \partial_x^2 u - 0.01\partial_x^3 u + 1.02\partial_x^4 u$	1.29	$u\partial_x u + \partial_x^2 u - 0.01\partial_x^3 u + 1.00\partial_x^4 u$	0.55

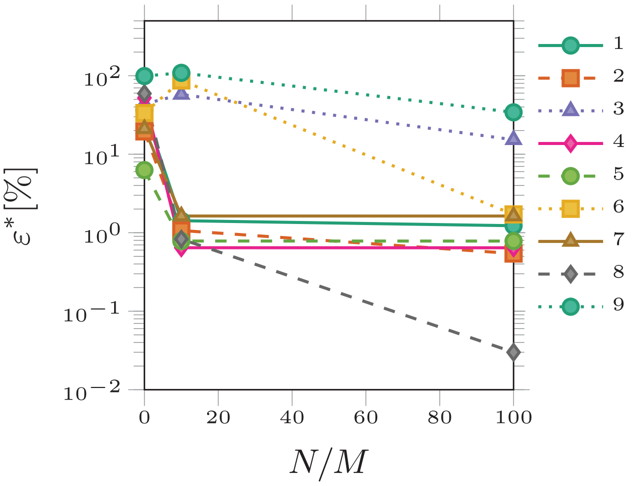


FIG. 2. Improvement in the performance of MEDIDA as the ensemble size, N, is increased ($n = 100$). $N/M = 0$ corresponds to no DA. ε^{*} for cases 3 and 9 further drops to 0.98% and 0.55% using larger ensembles of $N = 200M$ and $N = 500M$, respectively (Table I).

σ_u is the standard deviation of the state (equivalent to 0.1 following the methodology in Refs. 43 and 65). Without step 3 (DA), the model error discovery fails, leading to many spurious terms and ε^{*} of $O(10\%)$–$O(100\%)$, comparable or even worse than the model error, ε_m (Fig. 2).

Table I shows the corrected models (with steps 1–3). With $N = 10M$, in all cases except for 3, 6, and 9, $\varepsilon^{*} < 2\%$, between 30 and 60 times lower than ε_m. For those three cases, increasing N further reduces ε^{*} (Fig. 2), leading to the discovery of accurate models with $\varepsilon^{*} < 1\%$ at large ensembles with $N = 100M - 500M$ (Table I). Note that one common aspect of these cases is that the model is missing or mis-representing the hyper-diffusion term. The larger N is needed to further reduce the noise in the analysis state to prevent amplification of the noise due to the higher-order derivative of this term. It should also be highlighted that while N at the order of M or larger might seem impractical, each ensemble member requires only *one* time-step integration (thus, a total of nN time steps).

MEDIDA is found to be data-efficient in these experiments too. Like before, ε^{*} changes by 0.1% for $n \in \{10, 100, 1000\}$ for all cases except for 3, 6, and 9. For these three cases, ε^{*} improves by about 10% as n is increased from 10 to 100 but then changes by only 0.1% when n is further increased to 1000.

V. DISCUSSION AND SUMMARY

We introduced MEDIDA, a data-efficient and non-intrusive framework to discover interpretable, closed-form structural (or parametric) model error for chaotic systems. MEDIDA only needs a working numerical solver of the model and n sporadic pairs of noise-free or noisy observations of the system. Model error is estimated from differences between the observed and predicted states (obtained from one-time-step integration of the model from earlier observations). A closed-form representation of the model error is estimated using an equation-discovery technique such as RVM. MEDIDA is expected to work accurately if (i) the aforementioned differences are dominated by model error and not by numerical (discretization) error or observation noise and (ii) the library of the bases is adequate (further discussed below). Regarding (i), the numerical error can be reduced by using higher numerical resolutions while the observation noise can be tackled using DA techniques such as EnKF.

The performance of MEDIDA is demonstrated using a chaotic KS system for both noise-free and noisy observations. In the absence of noise, even with a small n, MEDIDA shows excellent performance and accurately discovers linear and nonlinear model errors, leading to corrected models that are very close to the system. These results already show the potential of MEDIDA as in some important applications, noise-free estimates of the system (reanalysis states) are routinely generated and are available; this is often the case for atmosphere and some other components of the Earth system.[66–68] For example, such reanalysis states have been recently used to correct the model errors of an NWP model,[28] a quasi-geostrophic (QG) model,[31] and a modular arbitrary-order-ocean-atmosphere model (MAOOAM).[69]

In the presence of noise, the model error could not be accurately discovered without DA. Once EnKF is employed, MEDIDA accurately discovers linear and nonlinear model errors. A few cases with model errors involving linear but fourth-order derivatives require larger ensembles. This is because higher-quality analysis states are needed to avoid amplification of any remaining noise as a result of high-order derivatives.

Although the number of ensemble members used in MEDIDA is more than what it is commonly used in operational DA, the associated cost is tractable since each ensemble is evolved only for one time step. This is another advantage of evolving the model for only one time step, which as discussed before, and is also motivated by reducing the accumulation of truncation errors and nonlinear growth of the model errors. Still, if needed, inexpensive surrogates of the model that could provide accurate one-time-step forecasts can be used to efficiently produce the ensembles.[35,70]

Also, note that in EnKF, ensemble members are evolved using the model. If the model error is large, the analysis state might be too far from the system's state, degrading the performance of MEDIDA. In all cases examined here, even though the model errors are large, with large-enough ensembles, the approximated analysis states are accurate enough to enable MEDIDA to discover the missing or inaccurate terms. However, in more complex systems, this might become a challenge that requires devising new remedies. One potential resolution is an iterative procedure, in which the analysis state is generated and a corrected model is identified, which is then used to produce a new analysis state and a new corrected model, and this continues until convergence. Although such iterative approaches currently lack proof of convergence, they have shown promises in similar settings;[39,58,71,72] for example, Hamilton *et al.*[58] update the error in the observations iteratively until the convergence of the estimated state from the EnKF. The corresponding cost and possible convergence properties of such an iterative approach for MEDIDA remain to be investigated. Similarly, other methods for dealing with noise in equation discovery[43,56,73] could also be integrated into MEDIDA.

The choice of an adequate library for RVM is crucial in MEDIDA. Although an exhaustive library of the training vectors with any arbitrary bases is straightforward, it quickly becomes computationally intractable. Any *a priori* knowledge of the system, such as locality, homogeneity,[30] Galilean invariance, and conservation properties,[41] can be considered to construct a concise library. Conversely, the library can be expanded to "explore the computational universe," e.g., using gene expression programming.[74] Even further, additional constraints, such as stability of the corrected model, can be imposed.[75–77] Effective strategies for the selection of an adequate and concise library can be investigated in future work using more complex test cases.

Finally, beyond dealing with noisy observations, another challenge in the discovery of interpretable model errors is approximating the library given spatially sparse observations. Approaches that leverage auto-differentiation for building the library in equation discovery have recently shown promising results[78] and can be integrated into MEDIDA.

MEDIDA is shown to work well for a widely used but simple chaotic prototype, the KS system. The next step will be investigating the performance of MEDIDA for more complex test cases, such as a two-layer quasi-geostrophic model. Scaling MEDIDA to more complex, higher dimensional systems will enable us to discover interpretable model errors for GCMs, NWP models, and other models of the climate system.

ACKNOWLEDGMENTS

We are grateful to Yifei Guan for helpful comments on the manuscript. We also would like to thank two anonymous reviewers for their constructive and insightful comments. This work was supported by an award from the ONR Young Investigator Program (No. N00014-20-1-2722), a grant from the NSF CSSI Program (No. OAC-2005123), and by the generosity of Eric and Wendy Schmidt by recommendation of the Schmidt Futures program. Computational resources were provided by NSF XSEDE (Allocation No. ATM170020) and NCAR's CISL (Allocation No. URIC0004).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Rambod Mojgani: Conceptualization (equal); Investigation (equal); Methodology (equal); Writing – original draft (equal); Writing – review & editing (equal). **Ashesh Chattopadhyay:** Conceptualization (equal); Methodology (equal); Writing – review and editing (equal). **Pedram Hassanzadeh:** Conceptualization (equal); Funding acquisition (lead); Resources (lead); Writing – review and editing (equal).

DATA AVAILABILITY

The code and data that support the findings of this study are available at https://github.com/envfluids/MEDIDA.

REFERENCES

[1] 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**, 1–17 (2020).

[2] F. Regazzoni, D. Chapelle, and P. Moireau, "Combining data assimilation and machine learning to build data-driven models for unknown long time dynamics—Applications in cardiovascular modeling," Int. J. Numer. Methods Biomed. Eng. **37**, e3471 (2021).

[3] K. E. Willcox, O. Ghattas, and P. Heimbach, "The imperative of physics-based modeling and inverse theory in computational science," Nat. Comput. Sci. **1**, 166–168 (2021).

[4] A. Subramanian and S. Mahadevan, "Model error propagation in coupled multiphysics systems," AIAA J. **58**, 2236–2245 (2020).

[5] K. Duraisamy, "Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence," Phys. Rev. Fluids **6**, 050504 (2021).

[6] V. Balaji, "Climbing down Charney's ladder: Machine learning and the post-Dennard era of computational climate science," Phil. Trans. R. Soc. A **379**, 20200085 (2021).

[7] T. Schneider, N. Jeevanjee, and R. Socolow, "Accelerating progress in climate science," Phys. Today **74**, 44–51 (2021).

[8] M. E. Levine and A. M. Stuart, "A framework for machine learning of model error in dynamical systems," arXiv:2107.06658 (2021).

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

[10] P. R. Vlachas, W. Byeon, Z. Y. Wan, T. P. Sapsis, and P. Koumoutsakos, "Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks," Proc. R. Soc. A **474**, 20170844 (2018).

[11] P. D. Dueben and P. Bauer, "Challenges and design choices for global weather and climate models based on machine learning," Geosci. Model. Dev. **11**, 3999–4009 (2018).

[12] J. A. Weyn, D. R. Durran, and R. Caruana, "Can machines learn to predict weather? Using deep learning to predict gridded 500-hPa geopotential height from historical weather data," J. Adv. Model. Earth Syst. **11**, 2680–2693 (2019).

[13] A. Chattopadhyay, P. Hassanzadeh, and D. Subramanian, "Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: Reservoir computing, artificial neural network, and long short-term memory network," Nonlinear Process. Geophys. **27**, 373–389 (2020).

[14] T. Arcomano, I. Szunyogh, J. Pathak, A. Wikner, B. R. Hunt, and E. Ott, "A machine learning-based global atmospheric forecast model," Geophys. Res. Lett. **47**, e2020GL087776, https://doi.org/10.1029/2020GL087776 (2020).

[15] A. Chattopadhyay, E. Nabizadeh, and P. Hassanzadeh, "Analog forecasting of extreme-causing weather patterns using deep learning," J. Adv. Model. Earth Syst. **12**, e2019MS001958 (2020).

[16] C. Ma, J. Wang, and E. Weinan, "Model reduction with memory and the machine learning of dynamical systems," arXiv:1808.04258 (2018).

[17] S. Rasp, M. S. Pritchard, and P. Gentine, "Deep learning to represent subgrid processes in climate models," Proc. Natl. Acad. Sci. U.S.A. **115**, 9684–9689 (2018).

[18] R. Maulik, O. San, A. Rasheed, and P. Vedula, "Subgrid modelling for two-dimensional turbulence using neural networks," J. Fluid Mech. **858**, 122–144 (2019).

[19] N. D. Brenowitz and C. S. Bretherton, "Spatially extended tests of a neural network parametrization trained by coarse-graining," J. Adv. Model. Earth Syst. **11**, 2728–2744 (2019).

[20] T. Bolton and L. Zanna, "Applications of deep learning to ocean data inference and subgrid parameterization," J. Adv. Model. Earth Syst. **11**, 376–399 (2019).

[21] A. Beck, D. Flad, and C.-D. Munz, "Deep neural networks for data-driven LES closure models," J. Comput. Phys. **398**, 108910 (2019).

[22] A. Subel, A. Chattopadhyay, Y. Guan, and P. Hassanzadeh, "Data-driven subgrid-scale modeling of forced Burgers turbulence using deep learning with generalization to higher Reynolds numbers via transfer learning," Phys. Fluids **33**, 031702 (2021).

[23] J. Harlim, S. W. Jiang, S. Liang, and H. Yang, "Machine learning for prediction with missing dynamics," J. Comput. Phys. **428**, 109922 (2021).

[24] Y. Guan, A. Chattopadhyay, A. Subel, and P. Hassanzadeh, "Stable a posteriori LES of 2D turbulence using convolutional neural networks: Backscattering analysis and generalization to higher *Re* via transfer learning," arXiv:2102.11400v1 (2021).

[25] P. A. Watson, "Applying machine learning to improve simulations of a chaotic dynamical system using empirical error correction," J. Adv. Model. Earth Syst. **11**, 1402–1417 (2019).

[26] S. Pawar, S. E. Ahmed, O. San, A. Rasheed, and I. M. Navon, "Long short-term memory embedded nudging schemes for nonlinear data assimilation of geophysical flows," Phys. Fluids **32**, 076606 (2020).

[27] J. Pathak, M. Mustafa, K. Kashinath, E. Motheau, T. Kurth, and M. Day, "Using machine learning to augment coarse-grid computational fluid dynamics simulations," arXiv:2010.00072 (2020).

[28] O. Watt-Meyer, N. D. Brenowitz, S. K. Clark, B. Henn, A. Kwa, J. McGibbon, W. A. Perkins, and C. S. Bretherton, "Correcting weather and climate models by machine learning nudged historical simulations," Geophys. Res. Lett. **48**, e2021GL092555, https://doi.org/10.1029/2021GL092555 (2021).

[29] C. S. Bretherton, B. Henn, A. Kwa, N. D. Brenowitz, O. Watt-Meyer, J. McGibbon, W. A. Perkins, S. K. Clark, and L. Harris, "Correcting coarse-grid weather and climate models by machine learning from global storm-resolving simulations," J. Adv. Model. Earth Syst. **14**, e2021MS002794 (2021).

[30] M. Bocquet, J. Brajard, A. Carrassi, and L. Bertino, "Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models," Nonlinear Process. Geophys. **26**, 143–162 (2019).

[31] A. Farchi, P. Laloyaux, M. Bonavita, and M. Bocquet, "Using machine learning to correct model error in data assimilation and forecast applications," Q. J. R. Meteorol. Soc. **147**, 3067–3084 (2021).

[32] J. Brajard, A. Carrassi, M. Bocquet, and L. Bertino, "Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the Lorenz 96 model," J. Comput. Sci. **44**, 101171 (2020).

[33] N. Chen and Y. Li, "BAMCAFE: A Bayesian machine learning advanced forecast ensemble method for complex nonlinear turbulent systems with partial observations," arXiv:2107.05549 (2021).

[34] A. Wikner, J. Pathak, B. R. Hunt, I. Szunyogh, M. Girvan, and E. Ott, "Using data assimilation to train a hybrid forecast system that combines machine-learning and knowledge-based components," Chaos **31**, 053114 (2021).

[35] A. Chattopadhyay, M. Mustafa, P. Hassanzadeh, E. Bach, and K. Kashinath, "Towards physically consistent data-driven weather forecasting: Integrating data assimilation with equivariance-preserving spatial transformers in a case study with ERA5," Geosci. Model Dev. Discuss. **15**, 1–23 (2021).

[36] G. A. Gottwald and S. Reich, "Combining machine learning and data assimilation to forecast dynamical systems from noisy partial observations," Chaos **31**, 101103 (2021).

[37] A. Chattopadhyay, A. Subel, and P. Hassanzadeh, "Data-driven super-parameterization using deep learning: Experimentation with multi-scale Lorenz 96 systems and transfer-learning," J. Adv. Model. Earth Syst. **12**, e2020MS002084 (2020).

[38] R. Mojgani and M. Balajewicz, "Low-rank registration based manifolds for convection-dominated PDEs," Proc. AAAI Conf. Artif. Intell. **35**, 399–407 (2021).

[39] 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. U.S.A. **113**, 3932–3937 (2016).

[40] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," Sci. Adv. **3**, e1602614 (2017).

[41] S. Zhang and G. Lin, "Robust data-driven discovery of governing physical laws with error bars," Proc. R. Soc. A **474**, 20180305 (2018).

[42] H. Schaeffer, G. Tran, and R. Ward, "Extracting sparse high-dimensional dynamics from limited data," SIAM J. Appl. Math. **78**, 3279–3295 (2018).

[43] P. A. K. 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).

[44] L. Zanna and T. Bolton, "Data-driven equation discovery of ocean mesoscale closures," Geophys. Res. Lett. **47**, e2020GL088376, https://doi.org/10.1029/2020GL088376 (2020).

[45] D. A. Messenger and D. M. Bortz, "Weak SINDy for partial differential equations," J. Comput. Phys. **443**, 110525 (2021).

[46] A. Cortiella, K. C. Park, and A. Doostan, "Sparse identification of nonlinear dynamical systems via reweighted $l1$-regularized least squares," Comput. Methods Appl. Mech. Eng. **376**, 113620 (2021).

[47] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," J. Mach. Learn. Res. **1**, 211–244 (2001).

[48] M. Asch, M. Bocquet, and M. Nodet, *Data Assimilation: Methods, Algorithms, and Applications* (SIAM, 2016).

[49] K. Law, A. Stuart, and K. Zygalakis, *Data Assimilation: A Mathematical Introduction*, Texts in Applied Mathematics (Springer International Publishing, 2015).

[50] C. E. Leith, "Objective methods for weather prediction," Annu. Rev. Fluid Mech. **10**, 107–128 (1978).

[51] A. Carrassi and S. Vannitsem, "Treatment of the error due to unresolved scales in sequential data assimilation," Int. J. Bifurcation Chaos **21**, 3619–3626 (2011).

[52] L. Mitchell and A. Carrassi, "Accounting for model error due to unresolved scales within ensemble Kalman filtering," Q. J. R. Meteorol. Soc. **141**, 1417–1428 (2015).

[53] S. Zhang and G. Lin, "SubTSBR to tackle high noise and outliers for data-driven discovery of differential equations," J. Comput. Phys. **428**, 109962 (2021).

[54] S. H. Rudy and T. P. Sapsis, "Sparse methods for automatic relevance determination," Physica D **418**, 132843 (2021).

[55] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, "Model selection for dynamical systems via sparse regression and information criteria," Proc. R. Soc. A **473**, 20170009 (2017).

[56] P. Goyal and P. Benner, "Discovery of nonlinear dynamical systems using a Runge-Kutta inspired dictionary-based sparse regression approach," arXiv:2105.04869 (2021).

[57] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," J. Geophys. Res.: Oceans **99**, 10143–10162, https://doi.org/10.1029/94JC00572 (1994).

[58] F. Hamilton, T. Berry, and T. Sauer, "Correcting observation model error in data assimilation," Chaos **29**, 053102 (2019).

[59] J. L. Anderson and S. L. Anderson, "A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts," Mon. Weather Rev. **127**, 2741–2758 (1999).

[60] A. Carrassi, M. Bocquet, L. Bertino, and G. Evensen, "Data assimilation in the geosciences: An overview of methods, issues, and perspectives," WIREs Climate Change **9**, e535 (2018).

[61] G. Evensen and P. J. van Leeuwen, "An ensemble Kalman smoother for nonlinear dynamics," Mon. Weather Rev. **128**, 1852–1867 (2000).

[62] M. Khodkar and P. Hassanzadeh, "A data-driven, physics-informed framework for forecasting the spatiotemporal evolution of chaotic dynamics with nonlinearities modeled as exogenous forcings," J. Comput. Phys. **440**, 110412 (2021).

[63] A.-K. Kassam and L. N. Trefethen, "Fourth-order time-stepping for stiff PDEs," SIAM J. Sci. Comput. **26**, 1214–1233 (2005).

[64] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006); available at https://link.springer.com/book/9780387310732.

[65] K. Kaheman, S. L. Brunton, and J. N. Kutz, "Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data," arXiv:2009.08810 (2020).

[66] J. Muñoz Sabater, E. Dutra, A. Agustí-Panareda, C. Albergel, G. Arduini, G. Balsamo, S. Boussetta, M. Choulga, S. Harrigan, H. Hersbach, B. Martens, D. G. Miralles, M. Piles, N. J. Rodríguez-Fernández, E. Zsoter, C. Buontempo, and J.-N. Thépaut, "ERA5-Land: A state-of-the-art global reanalysis dataset for land applications," Earth Syst. Sci. Data **13**, 4349–4383 (2021).

[67] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, A. Simmons, C. Soci, S. Abdalla, X. Abellan, G. Balsamo, P. Bechtold, G. Biavati, J. Bidlot, M. Bonavita, G. De Chiara, P. Dahlgren, D. Dee, M. Diamantakis, R. Dragani, J. Flemming, R. Forbes, M. Fuentes, A. Geer, L. Haimberger, S. Healy, R. J. Hogan, E. Hólm, M. Janisková, S. Keeley, P. Laloyaux, P. Lopez, C. Lupu, G. Radnoti, P. de Rosnay, I. Rozum, F. Vamborg, S. Villaume, and J.-N. Thépaut, "The ERA5 global reanalysis," Q. J. R. Meteorol. Soc. **146**, 1999–2049 (2020).

[68] Y. Fan, H. M. V. den Dool, D. Lohmann, and K. Mitchell, "1948–98 U.S. hydrological reanalysis by the Noah land data assimilation system," J. Clim. **19**, 1214–1237 (2006).
[69] J. Brajard, A. Carrassi, M. Bocquet, and L. Bertino, "Combining data assimilation and machine learning to infer unresolved scale parametrization," Phil. Trans. R. Soc. A **379**, 20200086 (2021).
[70] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, P. Hassanzadeh, K. Kashinath, and A. Anandkumar, "FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators," arXiv:2202.11214 (2022).
[71] H. Schaeffer, "Learning partial differential equations via data discovery and sparse optimization," Proc. R. Soc. A **473**, 20160446 (2017).
[72] T. Schneider, A. M. Stuart, and J.-L. Wu, "Ensemble Kalman inversion for sparse learning of dynamical systems from time-averaged data," arXiv:2007.06175 (2020).
[73] 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).
[74] H. Vaddireddy, A. Rasheed, A. E. Staples, and O. San, "Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensor observation data," Phys. Fluids **32**, 015113 (2020).
[75] R. Mojgani and M. Balajewicz, "Stabilization of linear time-varying reduced-order models: A feedback controller approach," Int. J. Numer. Methods Eng. **121**, 5490–5510 (2020).
[76] J. C. Loiseau and S. L. Brunton, "Constrained sparse Galerkin regression," J. Fluid Mech. **838**, 42–67 (2018).
[77] N. Chen, "Learning nonlinear turbulent dynamics from partial observations via analytically solvable conditional statistics," J. Comput. Phys. **418**, 109635 (2020).
[78] Z. Chen, Y. Liu, and H. Sun, "Physics-informed learning of governing equations from scarce data," Nat. Commun. **12**, 6136 (2021).