Journal of Computational Physics 418 (2020) 109635

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Learning nonlinear turbulent dynamics from partial observations via analytically solvable conditional statistics

Nan Chen

Department of Mathematics, University of Wisconsin-Madison, Madison, WI, USA

ARTICLE INFO

Article history:
Received 5 February 2020
Received in revised form 2 June 2020
Accepted 2 June 2020
Available online 16 June 2020

Keywords:
Expectation-maximization approach
Nonlinear optimal smoother
Short training data
Physics constraint
Block decomposition
Sparse identification

ABSTRACT

Learning nonlinear turbulent dynamics from partial observations is an important and challenging topic. In this article, an efficient learning algorithm based on the expectation-maximization approach is developed for a rich class of complex nonlinear turbulent dynamics using short training data. Despite the significant nonlinear and non-Gaussian features in these models, the analytically solvable conditional statistics allows the development of an exact and accurate nonlinear optimal smoother for recovering the hidden variables, which facilitates an efficient learning of these fully nonlinear models with extreme events. Then three additional ingredients are incorporated into the basic algorithm for improving the learning process. First, the physics constraint that requires the conservation of energy in the quadratic nonlinear terms is taken into account. It plays an important role in preventing the finite-time blowup of the solution and various pathological behavior of the recovered model. Second, a judicious block decomposition is applied to many large-dimensional nonlinear systems. It greatly accelerates the calculation of high-dimensional conditional covariance matrix and provides an extremely cheap parallel computation for learning the model parameters. Third, sparse identification of the complex turbulent models is combined with the learning algorithm that leads to parsimonious models. Numerical tests show the skill of the algorithm in learning the nonlinear dynamics and non-Gaussian statistics with extreme events in both perfect model and model error scenarios. It is also shown that in the presence of noise and partial observations, the model is not uniquely identified. Different nonlinear models all perfectly capture the key non-Gaussian features and obtain the same ensemble forecast skill of the observed variables as the perfect model, but they may have distinct model responses to external perturbations.

1. Introduction

Learning nonlinear turbulent dynamics from partial observations is an important topic in geophysics, engineering, neuroscience and material science [1–4]. In many situations, the exact parameter values, or even the gross structure of the dynamical system itself, are unknown. The dynamics of the system have to be learned or identified from partial observations. Such a learning process plays a crucial role in discovering the underlying physics. It is also a necessary precursor for effective state estimation, data assimilation and prediction [5–9]. In addition to identifying strong nonlinear interactions

E-mail address: chennan@math.wisc.edu.

https://doi.org/10.1016/j.jcp.2020.109635
0021-9991/

between different variables, significant non-Gaussian features such as fat-tailed probability density function (PDF), intermittency and extreme events [10–12], which result from both the nonlinearity and multiplicative noise, are expected to be reproduced in the recovered nonlinear dynamics. Unfortunately, the noisy partial observations often prevent efficient learning of the model structure and parameters for systems involving strong nonlinearity. In addition, only very limited training data is available in many real applications including the climate, atmosphere and ocean science, which brings about extra difficulties in learning the underlying dynamics and leads to the failure of most purely data-driven non-parametric methods. Furthermore, high dimensionality of both the state variable phase space and parameter space is a common feature of many complex dynamical systems. Developing efficient methods for high-dimensional systems to reduce computational cost is thus crucial in identifying the underlying dynamics. Finally, learning the model parameters and structure in the presence of model error [13–17] needs to be taken into account in many practical issues.

Learning nonlinear turbulent dynamics from partial observations can be linked to an optimization problem, where the parameters or the model structure are determined by optimizing a certain objective function. For simple systems with linear structure and Gaussian noise, the closed forms of the likelihood or other objective functions allow the model to be efficiently and accurately identified [18,19]. However, due to the intrinsic nonlinearity, non-Gaussian features, and partial observations in many complex physical problems, closed analytic formulae are typically not available. To this end, many numerical or approximate methods have been developed for parameter estimation and model identification. Markov chain Monte Carlo (MCMC) method is a commonly used approach that is applied to many complex nonlinear dynamical systems [20–24]. In the presence of partial or incomplete observations, the MCMC algorithm is often combined with data augmentation [25–27] to sample the missing trajectories and parameters simultaneously. In particular, the so-called pseudo-marginal approach can be applied to accelerate the MCMC parameter estimation methods by retaining a reasonable acceptance rate [28,29]. On the other hand, regarding the model parameters as augmented state variables, ensemble Kalman filter and particle filter can be applied for online parameter estimation [30–32]. In addition, finding the solutions associated with the maximum a posteriori or maximum likelihood estimates [33,34,30] with certain numerical approximations is also a widely used method in coping with many nonlinear problems.

Despite the success and recent advance of these sampling or numerical methods in many applications, incorporating closed analytical formulae into the calculation of the objective function is still highly preferable for learning complex nonlinear dynamics with strong non-Gaussian features. In particular, the computational advantage of the closed analytical formulae over various numerical methods becomes significant when the dimension of the underlying system increases and the non-Gaussian features become stronger. Notably, in the presence of partial observations, recovering the hidden process based on closed analytical formulae is computationally much cheaper and more accurate compared with sampling the missing trajectories numerically. The closed analytical formulae also facilitate the theoretic study of the learning algorithms.

Since the perfect model is typically unknown or it is too complicated to use in practice, model identification and parameter estimation are often based on specific classes of approximate models. Therefore, from a practical point of view, it is of significance to develop a suitable nonlinear modeling framework, which includes a rich class of nonlinear turbulent dynamical systems with key non-Gaussian features as observed in nature. Meanwhile, these nonlinear turbulent models involve certain analytically solvable statistics that facilitate the process of learning the model structure and parameters. These nonlinear models then serve as suitable approximate or surrogate models for describing and forecasting nature.

In a recent work [35,36], a conditional Gaussian nonlinear modeling framework was developed. These systems are highly nonlinear and non-Gaussian, where both the joint and marginal PDFs can be skewed with fat tails. Extreme events, intermittency and highly nontrivial nonlinear interactions between different variables all appear in the conditional Gaussian systems. The name 'conditional Gaussian' comes from the fact that once the trajectories of a subset of the variables are known, the statistics of the remaining variables conditioned on the given trajectories are Gaussian. The conditional Gaussian modeling framework includes a large class of the physics-constrained nonlinear stochastic models [37,38], many stochastically coupled reaction-diffusion models in neuroscience and ecology [39,40], and quite a few important large-scale dynamical models in turbulence, fluids and geophysical flows [41,42]. A gallery of examples of conditional Gaussian systems can be found in [35]. One important feature of the conditional Gaussian nonlinear models is that the conditional statistics are given by closed analytic formulae, which provide an efficient approach for learning the model structure and parameters.

In this article, an efficient algorithm based on the expectation-maximization (EM) approach [43,18,19] is developed for the model identification and parameter estimation of the conditional Gaussian nonlinear systems with partial observations. Despite the strong nonlinearity, the explicit formulae of the conditional statistics facilitate a closed form of the so-called nonlinear optimal smoother [44], which plays a key role in efficiently computing the nonlinear optimal state estimation of the unobserved variables in the E-Step. It also advances an explicit formula of calculating the likelihood in the M-Step. One salient advantage of the algorithm is that it only requires a short training data in learning the significant nonlinear and non-Gaussian features of the underlying dynamics even in the presence of small-scale noise. Note that without the closed analytical formulae for the nonlinear optimal smoother, numerical and approximate methods have to be used in the E-Step [45,18], which typically involve a local linearization of the system such that the extended Kalman smoother (EKS) can be applied. However, such a linear approximation often leads to large errors for nonlinear systems with strong intermittency and extreme events. In addition, some non-parametric approximations are often adopted to facilitate the application of the EKS, which however may introduce extra errors and prevent the use of the algorithm for high-dimensional systems.

Then several additional ingredients are incorporated into the basic EM algorithm for improving the learning of the nonlinear dynamics. First, the conservation of energy in the quadratic nonlinear terms, known as the physics constraint [37,38],

is a key feature in the development of nonlinear turbulent dynamical systems. The physics constraint prevents finite time blowup of statistical solutions and pathological behavior of their invariant measure [46]. Incorporating the physics constraint (as well as some other constraints) into the basic learning algorithm can be achieved by constructing a Lagrangian function in the M-Step. Such a manipulation still guarantees using closed analytic formulae for efficiently optimizing the likelihood function. Second, many large-dimensional systems with multiscale structures [47], multilevel dynamics [48] or state-dependent parameterizations [49] have localized covariance structures associated with different state variables [50]. Such a feature allows applying a judicious block decomposition [51,52] to the full system that greatly accelerates the calculation of high-dimensional conditional covariance matrix resulting from the nonlinear smoother in the E-Step. It also facilitates solving the likelihood function and provides an extremely cheap parallel computation for estimating different parameters. Third, sparse identification of the complex turbulent models is often preferred in order to prevent overfitting and provide a parsimonious model [53–55]. The least absolute shrinkage and selection operator (LASSO) [56,57] can be easily incorporated into the basic learning algorithm, which also advances the study of the uniqueness and the predictability of the identified models from partial observations.

The rest of the article is organized as follows. The general conditional Gaussian nonlinear modeling framework is introduced in Section 2, which is followed by the closed analytical formulae of the optimal nonlinear smoother estimates. The basic efficient learning algorithm applying to the conditional Gaussian nonlinear models based on the EM approach is developed in Section 3. Section 4 focuses on the improved algorithms, including incorporating the physics constraint and the sparse model identification into the basic algorithm, and the development of the block decomposition technique for high-dimensional systems. The quantification of the skill in learning the model and parameters is discussed in Section 5. Examples of learning model parameters with significant non-Gaussian features using short training data are shown in Section 6, where both the perfect model tests and those with model error are included. The role of imposing the physics constraint in the learning process is also highlighted in this section. Section 7 studies learning the model from partial observations with unknown model structure. The article is concluded in Section 8.

2. The conditional Gaussian nonlinear models

2.1. The modeling framework

The conditional Gaussian nonlinear models have the following general form [36,35],

$$d\mathbf{X}(t) = \Big[\mathbf{A_0}(\mathbf{X},t) + \mathbf{A_1}(\mathbf{X},t)\mathbf{Y}(t)\Big]\,dt + \mathbf{B_X}(\mathbf{X},t)\,d\mathbf{W_1}(t), \tag{1a}$$

$$d\mathbf{Y}(t) = \Big[\mathbf{a_0}(\mathbf{X},t) + \mathbf{a_1}(\mathbf{X},t)\mathbf{Y}(t)\Big]\,dt + \mathbf{b_Y}(\mathbf{X},t)\,d\mathbf{W_2}(t), \tag{1b}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are both multi-dimensional state variables. In (1), $\mathbf{A}_0$, $\mathbf{A}_1$, $\mathbf{a_0}$, $\mathbf{a_1}$, $\mathbf{B_X}$ and $\mathbf{b_Y}$ and are vectors and matrices that depend nonlinearly on the state variables $\mathbf{X}$ and time t, and they may also contain parameters, e.g., $\mathbf{A_0}(\mathbf{X},t) := \mathbf{A_0}(\mathbf{X},t;\boldsymbol{\theta})$ while $\mathbf{W_1}$ and $\mathbf{W_2}$ are independent white noise. For nonlinear systems with partial observations, $\mathbf{X}$ can be regarded as the collection of the observed variables while $\mathbf{Y}$ contains the variables that are not directly observed.

The systems in (1) are called conditional Gaussian nonlinear systems, because given a realization of $\mathbf{X}(s)$ up to time $s = t$ the distribution of $\mathbf{Y}(t)$ conditioned on such a realization, namely $p(\mathbf{Y}(t)|\mathbf{X}(s), s \le t)$, is Gaussian. Despite the conditional Gaussianity, the coupled system (1) remains highly nonlinear and is able to capture many non-Gaussian features as appear in nature.

The conditional Gaussian nonlinear modeling framework (1) includes many physics-constrained nonlinear stochastic models [37,38], large-scale dynamical models in turbulence, fluids and geophysical flows, as well as stochastically coupled reaction-diffusion models in neuroscience and ecology. Some concrete examples are noisy versions of the Lorenz systems (Lorenz 63, Lorenz 84 and a two layer Lorenz 96 models), a variety of the stochastically coupled FitzHugh-Nagumo model, and the Boussinesq equations with noise. See a recent work [35] for a gallery of examples of the conditional Gaussian systems. Applications of the conditional Gaussian systems to strongly nonlinear systems include developing low-order nonlinear stochastic models for predicting the non-Gaussian intermittent time series of the Madden-Julian oscillation (MJO) and the monsoon intraseasonal variabilities [58–61], filtering the stochastic skeleton model for the MJO [62], and recovering the turbulent ocean flows with noisy observations from Lagrangian tracers [63–65]. Other studies that also fit into the conditional Gaussian framework includes the cheap exactly solvable forecast models in dynamic stochastic superresolution of sparsely observed turbulent systems [49,66], stochastic superparameterization for geophysical turbulence [47] and blended particle filters for large-dimensional chaotic systems [67].

2.2. Nonlinear optimal filter and smoother

Although various numerical and approximate methods have to be used for state estimation of general nonlinear systems, the conditional Gaussian nonlinear modeling framework allows closed analytic formulae for solving the nonlinear optimal filter and nonlinear optimal smoother estimates. The state estimation based on such closed forms plays a key role in the development of efficient and accurate learning algorithms.

Theorem 2.1 *(Nonlinear optimal filter). Given one realization of the time series* $\mathbf{X}(s)$ *for* $s \in [0, t]$*, the conditional distribution*

$$p(\mathbf{Y}(t)|\mathbf{X}(s), s \le t) \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{f}(t), \mathbf{R}_\mathbf{f}(t)) \tag{2}$$

is Gaussian, where the conditional mean $\boldsymbol{\mu}_\mathbf{f}$ *and the conditional covariance* $\mathbf{R}_\mathbf{f}$ *are given by the following explicit formulae,*

$$\mathrm{d}\boldsymbol{\mu}_\mathbf{f} = (\mathbf{a_0} + \mathbf{a_1}\boldsymbol{\mu}_\mathbf{f})\,\mathrm{d}t + (\mathbf{R_f}\mathbf{A}_\mathbf{1}^*)(\mathbf{B_X}\mathbf{B}_\mathbf{X}^*)^{-1}(\,\mathrm{d}\mathbf{X} - (\mathbf{A_0} + \mathbf{A_1}\boldsymbol{\mu}_\mathbf{f})\,\mathrm{d}t), \tag{3a}$$

$$\mathrm{d}\mathbf{R}_\mathbf{f} = \big(\mathbf{a_1}\mathbf{R_f} + \mathbf{R_f}\mathbf{a}_\mathbf{1}^* + \mathbf{b_Y}\mathbf{b}_\mathbf{Y}^* - (\mathbf{R_f}\mathbf{A}_\mathbf{1}^*)(\mathbf{B_X}\mathbf{B}_\mathbf{X}^*)^{-1}(\mathbf{A_1}\mathbf{R_f})\big)\,\mathrm{d}t. \tag{3b}$$

See [68] for the proof of Theorem 2.1. The formulae in (3) are the nonlinear optimal filter estimates for the conditional Gaussian nonlinear systems, where the conditional covariance is driven by a random Riccati equation. The conditional mean $\boldsymbol{\mu}_\mathbf{f}$ and the conditional covariance $\mathbf{R}_\mathbf{f}$ in (3) are also named as posterior mean and posterior covariance or filter mean and filter covariance. The classical Kalman-Bucy filter [69] is the simplest example within the conditional Gaussian framework.

In addition to the optimal nonlinear filtering result, the conditional Gaussian framework also allows the development of closed analytic formulae for the nonlinear optimal smoother estimate. Note that the fundamental difference between filtering and smoothing lies in the observational data used for state estimation. In filtering, the observational data only up to the current time instant t is adopted and therefore the filtering technique is an online (or real-time) method. On the other hand, the data in the entire observational period $[0, T]$ is used in the smoothing technique. Thus, the smoother allows a more accurate and unbiased state estimation compared with the filter despite that the state estimation is no longer real time since the information beyond the current time instant is used in the state estimation as well. The details of deriving the nonlinear smoother have been illustrated in a recent work [44] and the key results are summarized as follows.

Theorem 2.2 *(Nonlinear optimal smoother). Given one realization of the observed variable* $\mathbf{X}(t)$ *for* $t \in [0, T]$*, the optimal smoother estimate* $p(\mathbf{Y}(t)|\mathbf{X}(s), s \in [0, T])$ *is conditional Gaussian,*

$$p(\mathbf{Y}(t)|\mathbf{X}(s), s \in [0, T]) \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{s}(t), \mathbf{R}_\mathbf{s}(t)), \tag{4}$$

where the conditional mean $\boldsymbol{\mu}_\mathbf{s}(t)$ *and conditional covariance* $\mathbf{R}_\mathbf{s}(t)$ *of the smoother at time t satisfy the following backward equations*

$$\overleftarrow{\mathrm{d}\boldsymbol{\mu}_\mathbf{s}} = \big(-\mathbf{a_0} - \mathbf{a_1}\boldsymbol{\mu}_\mathbf{s} + (\mathbf{b_Y}\mathbf{b}_\mathbf{Y}^*)\mathbf{R}_\mathbf{f}^{-1}(\boldsymbol{\mu}_\mathbf{f} - \boldsymbol{\mu}_\mathbf{s})\big)\,\mathrm{d}t, \tag{5a}$$

$$\overleftarrow{\mathrm{d}\mathbf{R}_\mathbf{s}} = -\big((\mathbf{a_1} + (\mathbf{b_Y}\mathbf{b}_\mathbf{Y}^*)\mathbf{R}_\mathbf{f}^{-1})\mathbf{R_s} + \mathbf{R_s}(\mathbf{a}_\mathbf{1}^* + (\mathbf{b_Y}\mathbf{b}_\mathbf{Y}^*)\mathbf{R}_\mathbf{f}^{-1}) - \mathbf{b_Y}\mathbf{b}_\mathbf{Y}^*\big)\,\mathrm{d}t, \tag{5b}$$

where $\boldsymbol{\mu}_\mathbf{f}$ *and* $\mathbf{R}_\mathbf{f}$ *are given by (3). In (5), the terms on the left hand side are understood as*

$$\overleftarrow{\mathrm{d}\boldsymbol{\mu}_\mathbf{s}} = \lim_{\Delta t \to 0} \boldsymbol{\mu}_\mathbf{s}(t) - \boldsymbol{\mu}_\mathbf{s}(t + \Delta t),$$

$$\overleftarrow{\mathrm{d}\mathbf{R}_\mathbf{s}} = \lim_{\Delta t \to 0} \mathbf{R}_\mathbf{s}(t) - \mathbf{R}_\mathbf{s}(t + \Delta t).$$

The starting value of the nonlinear smoother $(\boldsymbol{\mu}_\mathbf{s}(T), \mathbf{R}_\mathbf{s}(T))$ *is the same as the filter estimate at the endpoint* $(\boldsymbol{\mu}_\mathbf{f}(T), \mathbf{R}_\mathbf{f}(T))$*.*

The subscripts $\cdot_\mathbf{f}$ and $\cdot_\mathbf{s}$ stand for "filter" and "smoother" respectively.

3. The basic learning algorithm

Assume the coupled conditional Gaussian nonlinear system (1) is only partially observed, where $\mathbf{X}(s), s \in [0, T]$ is the observed time series that is typically not too long. For the simplicity of discussions, let us apply an Euler-Maruyama scheme [70,71] to the original continuous system (1) with a small but finite time step Δt. Thus, the values of $\mathbf{X}$ and $\mathbf{Y}$ are taken at discrete points in time $\mathbf{X}(t_j)$ and $\mathbf{Y}(t_j)$, for $j = 0, 1, \ldots, J$, where $T = J\Delta t$. Define $\widehat{\mathbf{X}} = \{\mathbf{X}^0, \ldots, \mathbf{X}^j, \ldots, \mathbf{X}^J\}$ and $\widehat{\mathbf{Y}} = \{\mathbf{Y}^0, \ldots, \mathbf{Y}^j, \ldots, \mathbf{Y}^J\}$, where $\mathbf{X}^j := \mathbf{X}(t_j)$ and $\mathbf{Y}^j = \mathbf{Y}(t_j)$.

Given an ansatz of the model, the goal here is to maximize the objective function, which is the log likelihood,

$$\mathcal{L}(\boldsymbol{\theta}) = \log p(\widehat{\mathbf{X}}|\boldsymbol{\theta}) = \log \int_{\widehat{\mathbf{Y}}} p(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}|\boldsymbol{\theta})\,\mathrm{d}\widehat{\mathbf{Y}}, \tag{6}$$

where $\boldsymbol{\theta}$ is the collection of model parameters.

3.1. The basic expectation-maximization (EM) algorithm

Using any distribution $Q(\widehat{\mathbf{Y}})$ over the hidden variables, a lower bound on the likelihood $\mathcal{L}$ can be obtained in the following way [45],

$$\begin{aligned}\log \int_{\widehat{\mathbf{Y}}} p(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}|\boldsymbol{\theta})\, \mathrm{d}\widehat{\mathbf{Y}} &= \log \int_{\widehat{\mathbf{Y}}} Q(\widehat{\mathbf{Y}}) \frac{p(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}|\boldsymbol{\theta})}{Q(\widehat{\mathbf{Y}})}\, \mathrm{d}\widehat{\mathbf{Y}} \\ &\geq \int_{\widehat{\mathbf{Y}}} Q(\widehat{\mathbf{Y}}) \log \frac{p(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}|\boldsymbol{\theta})}{Q(\widehat{\mathbf{Y}})}\, \mathrm{d}\widehat{\mathbf{Y}} \\ &= \int_{\widehat{\mathbf{Y}}} Q(\widehat{\mathbf{Y}}) \log p(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}|\boldsymbol{\theta})\, \mathrm{d}\widehat{\mathbf{Y}} - \int_{\widehat{\mathbf{Y}}} Q(\widehat{\mathbf{Y}}) \log Q(\widehat{\mathbf{Y}})\, \mathrm{d}\widehat{\mathbf{Y}} \\ &:= \mathcal{F}(Q, \boldsymbol{\theta}),\end{aligned} \tag{7}$$

where the negative value of $\int_{\widehat{\mathbf{Y}}} Q(\widehat{\mathbf{Y}}) \log p(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}|\boldsymbol{\theta})\, \mathrm{d}\widehat{\mathbf{Y}}$ is the so-called free energy while $-\int_{\widehat{\mathbf{Y}}} Q(\widehat{\mathbf{Y}}) \log Q(\widehat{\mathbf{Y}})\, \mathrm{d}\widehat{\mathbf{Y}}$ is the entropy. Therefore, based on the fact $\mathcal{F}(Q, \boldsymbol{\theta}) \leq \mathcal{L}(\boldsymbol{\theta})$, it is clear that maximizing the log likelihood is equivalent to maximizing $\mathcal{F}$ alternatively with respect to the distribution Q and the parameters $\boldsymbol{\theta}$. This can be achieved by the expectation-maximization (EM) algorithm [43,72,73],

$$\text{E-Step:} \qquad Q_{k+1} \leftarrow \arg\max_{Q} \mathcal{F}(Q, \boldsymbol{\theta}_k), \tag{8a}$$

$$\text{M-Step:} \qquad \boldsymbol{\theta}_{k+1} \leftarrow \arg\max_{\boldsymbol{\theta}} \mathcal{F}(Q_{k+1}, \boldsymbol{\theta}). \tag{8b}$$

The maximization in the E-Step is reached when Q is exactly the conditional distribution of $\widehat{\mathbf{Y}}$ corresponding to the smoother estimates, that is,

$$Q_{k+1}(\widehat{\mathbf{Y}}) = p(\widehat{\mathbf{Y}}|\widehat{\mathbf{X}}, \boldsymbol{\theta}_k). \tag{9}$$

In such a situation, the bound in (7) becomes an equality $\mathcal{F}(Q, \boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta})$. Note that the conditional distribution in the E-Step is very difficult to solve for general nonlinear systems. Various numerical methods and approximations are often used [45,18], which however may suffer from both the approximation errors and the curse of dimensionality. Nevertheless, for the conditional Gaussian systems, the distribution $p(\widehat{\mathbf{Y}}|\widehat{\mathbf{X}}, \boldsymbol{\theta}_k)$ is given by the closed analytic formulae of the nonlinear smoother in Theorem 2.2, which greatly facilitates the application of the EM algorithm to many nonlinear models.

On the other hand, since the entropy (the second term on the right hand side of (7)) does not depend on $\boldsymbol{\theta}$, the maximum in the M-Step is obtained by maximizing the negative of the free energy

$$\boldsymbol{\theta}_{k+1} \leftarrow \arg\max_{\boldsymbol{\theta}} \int_{\widehat{\mathbf{Y}}} p(\widehat{\mathbf{Y}}|\widehat{\mathbf{X}}, \boldsymbol{\theta}_k) \log p(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}|\boldsymbol{\theta})\, \mathrm{d}\widehat{\mathbf{Y}}. \tag{10}$$

3.2. Applying the EM algorithm to the conditional Gaussian nonlinear models

Let us start with writing down the discrete approximation of the original continuous system using the Euler-Maruyama scheme,

$$\begin{aligned}\mathbf{X}^{j+1} &= \mathbf{X}^j + (\mathbf{A}_{\mathbf{0}}^j + \mathbf{A}_{\mathbf{1}}^j \mathbf{Y}^j)\Delta t + (\mathbf{B}_{\mathbf{X}}^j \sqrt{\Delta t})\varepsilon_{\mathbf{X}}^j, \\ \mathbf{Y}^{j+1} &= \mathbf{Y}^j + (\mathbf{a}_{\mathbf{0}}^j + \mathbf{a}_{\mathbf{1}}^j \mathbf{Y}^j)\Delta t + (\mathbf{b}_{\mathbf{Y}}^j \sqrt{\Delta t})\varepsilon_{\mathbf{Y}}^j,\end{aligned} \tag{11}$$

where $\varepsilon_{\mathbf{X}}^j$ and $\varepsilon_{\mathbf{Y}}^j$ are standard independent and identically distributed Gaussian random variables. Assume all the parameters appear as a multiplicative prefactor of some functions of $\mathbf{X}^j$ and $\mathbf{Y}^j$ on the right hand side of (11), which after certain change of variables is valid for many nonlinear turbulent models. Then the log likelihood function of $p(\widehat{\mathbf{X}}, \widehat{\mathbf{Y}}|\boldsymbol{\theta})$ and in turn the M-Step (9) can be solved explicitly. Denote $\mathbf{Z}^j = (\mathbf{X}^j, \mathbf{Y}^j)^{\mathrm{T}}$. If $\mathbf{Z}$ is fully observed, then clearly the one-step log-likelihood function under the model (11) would be

$$\mathcal{N}(\boldsymbol{\mu}^j, \mathbf{R}^j) = \widetilde{C}|\mathbf{R}^j|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{Z}^{j+1} - \boldsymbol{\mu}^j)^*(\mathbf{R}^j)^{-1}(\mathbf{Z}^{j+1} - \boldsymbol{\mu}^j)\right), \tag{12}$$

where $\boldsymbol{\mu}^j = \mathbf{M}^j\boldsymbol{\theta} + \mathbf{C}^j$ and $\mathbf{C}^j$ are those terms that do not involve parameters such as the first terms on the right hand side of (11), namely $\mathbf{X}^j$ and $\mathbf{Y}^j$. The covariance $\mathbf{R}^j$ is a block diagonal matrix with entries $(\mathbf{B}_{\mathbf{X}}^j)^2\Delta t$ and $(\mathbf{b}_{\mathbf{Y}}^j)^2\Delta t$. The constant $\widetilde{C}$ is due to the normalization of a Gaussian distribution.

However, in the partially observed nonlinear systems considered here, the variable $\mathbf{Y}$ is unobserved. The states of $\mathbf{Y}$ are estimated from the nonlinear smoother (5), which means the estimation contains uncertainty. Thus, an expectation of the log-likelihood function as in (10) needs to be taken, and the overall objective function becomes

$$\min_{\boldsymbol{\theta},\mathbf{R}} \widetilde{\mathcal{L}} = \min_{\boldsymbol{\theta},\mathbf{R}} \Big(\sum_j \Big\langle (\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)^*(\mathbf{R})^{-1}(\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j) \Big\rangle + J \log|\mathbf{R}| \Big), \tag{13}$$

where $\langle\cdot\rangle$ denotes the expectation over the uncertain component of $\mathbf{Z}^j$, namely $\mathbf{Y}^j$, at fixed j while the expectation of the deterministic component $\mathbf{X}^j$ is simply itself. In (13), we have made use of the fact that maximizing the log likelihood is equivalent to minimizing the negative log likelihood. In addition, we have made a further assumption that $\mathbf{R}^j = \mathbf{R}$ for all j. This means the coefficients $\mathbf{B}_{\mathbf{X}}^j$ and $\mathbf{b}_{\mathbf{Y}}^j$ are constants. Note that this simplification is not necessary for the development of the algorithm here but it will make the derivations more concise. Once $\mathbf{R}$ is estimated, the noise coefficients are obtained. The total number of the parameters equals the summation of that in $\boldsymbol{\theta}$ and that in $\mathbf{R}$. To find the minimum of $\widetilde{\mathcal{L}}$, we aim at finding $\frac{\partial \widetilde{\mathcal{L}}}{\partial \boldsymbol{\theta}} = 0$ and $\frac{\partial \widetilde{\mathcal{L}}}{\partial \mathbf{R}} = 0$. Here

$$\begin{aligned} \widetilde{\mathcal{L}} &= -J \log|\mathbf{R}^{-1}| + \sum_j \Big\langle (\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)^*(\mathbf{R})^{-1}(\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j) \Big\rangle \\ &= -J \log|\mathbf{R}^{-1}| + \mathrm{Tr}\left(\sum_j \Big\langle \Big(\mathbf{R}^{-1}(\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)(\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)^* \Big) \Big\rangle \right), \end{aligned} \tag{14}$$

and therefore

$$\frac{1}{J}\frac{\partial \widetilde{\mathcal{L}}}{\partial(\mathbf{R}^{-1})} = -\mathbf{R} + \frac{1}{J}\sum_j \Big\langle (\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)(\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)^* \Big\rangle. \tag{15}$$

This gives

$$\mathbf{R} = \frac{1}{J}\sum_j \Big\langle (\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)(\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)^* \Big\rangle. \tag{16}$$

On the other hand,

$$\frac{\partial \widetilde{\mathcal{L}}}{\partial \boldsymbol{\theta}} = 2\sum_j \Big\langle (\mathbf{M}^j)^*\mathbf{R}^{-1}\mathbf{M}^j \Big\rangle \boldsymbol{\theta} - 2\sum_j \Big\langle (\mathbf{M}^j)^*\mathbf{R}^{-1}(\mathbf{Z}^{j+1} - \mathbf{C}^j) \Big\rangle = 0, \tag{17}$$

which implies

$$\boldsymbol{\theta} = \left(\sum_j \Big\langle (\mathbf{M}^j)^*\mathbf{R}^{-1}\mathbf{M}^j \Big\rangle \right)^{-1} \left(\sum_j \Big\langle (\mathbf{M}^j)^*\mathbf{R}^{-1}(\mathbf{Z}^{j+1} - \mathbf{C}^j) \Big\rangle \right). \tag{18}$$

In the derivations of (16) and (18), the following facts have been made,

$$\frac{\partial}{\partial \mathbf{R}}(\log|\mathbf{R}|) = \mathbf{R}^{-1} \qquad \text{and} \qquad \frac{\partial}{\partial \mathbf{R}}(\mathrm{Tr}(\mathbf{PR})) = \frac{\partial}{\partial \mathbf{R}}(\mathrm{Tr}(\mathbf{RP})) = \mathbf{P}^*, \tag{19}$$

where $\mathbf{P}$ is a matrix and Tr is the matrix trace.

The details of solving (16) and (18) for conditional Gaussian nonlinear systems are included in Appendix Sections A.1–A.2.

4. The improved algorithms

4.1. Learning parameters with physics constraint and other constraints

Taking into account prior knowledge from physics, observations or experiments facilitates the learning of many complex dynamical systems using partial observations. This can be achieved by imposing extra conditions or constraints in the learning process. In particular, one of the most important constraints in modeling nonlinear turbulent dynamical systems is the so-called physics constraint [38,37], which requires the energy in the quadratic nonlinear terms to be conserved. In fact, without taking into account the physics constraint, ad hoc quadratic multilevel regression models can have finite time blowup of statistical solutions and pathological behavior of their invariant measure even though they match the data with high precision [46]. Recently, a new class of physics-constrained multi-level nonlinear regression models was developed [37,38] which involve both memory effects in time as well as physics constrained energy conserving nonlinear interactions

and completely avoid the above pathological behavior with full mathematical rigor. It has been shown in [35] that many of these physics-constrained models have conditional Gaussian structures.

To this end, it is important to incorporate the physics constraint as well as other constraints into the parameter estimation framework. These constraints require the combinations of certain parameters to satisfy some equality relationships. Therefore, it is natural to propose the following general form of the constraints

$$\mathbf{H}\boldsymbol{\theta} = \mathbf{g}, \tag{20}$$

where $\mathbf{H}$ is a matrix with each row standing for one equality constraint while $\mathbf{g}$ is a column vector. Incorporating the constraint (20) into the basic learning framework (14) leads to the following constrained optimization problem

$$\begin{aligned} &\min_{\boldsymbol{\theta},\mathbf{R}} \widetilde{\mathcal{L}} = \min_{\boldsymbol{\theta},\mathbf{R}} \left(-\frac{J}{2}\log|\mathbf{R}^{-1}| + \frac{1}{2}\sum_j \left\langle (\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)^*(\mathbf{R})^{-1}(\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)\right\rangle \right) \\ &\text{s.t.} \quad \mathbf{H}\boldsymbol{\theta} = \mathbf{g}, \end{aligned} \tag{21}$$

which can be solved by using the Lagrangian multiplier method,

$$\begin{aligned} f(\boldsymbol{\theta},\mathbf{R},\boldsymbol{\lambda}) = &\frac{1}{2}\sum_j \left\langle (\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)^*(\mathbf{R})^{-1}(\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)\right\rangle \\ &- \frac{J}{2}\log|\mathbf{R}^{-1}| + \boldsymbol{\lambda}^*(\mathbf{H}\boldsymbol{\theta} - \mathbf{g}). \end{aligned} \tag{22}$$

The solution of (22) is given by the zeros of $\partial f/\partial\boldsymbol{\theta} = 0$, $\partial f/\partial\mathbf{R} = 0$, and $\partial f/\partial\boldsymbol{\lambda} = 0$. Since the constraints do not appear in the diffusion part, the solution of $\mathbf{R}$ remains the same as that in (16). On the other hand, the constraints modify the solution of $\boldsymbol{\theta}$, which is given by the following Karush Kuhn Tucker (KKT) equation [74],

$$\begin{aligned} &\frac{\partial f}{\partial\boldsymbol{\theta}} = \sum_j \left\langle (\mathbf{M}^j)^*\mathbf{R}^{-1}\mathbf{M}^j\right\rangle\boldsymbol{\theta} - \sum_j \left\langle (\mathbf{M}^j)^*\mathbf{R}^{-1}(\mathbf{Z}^{j+1} - \mathbf{C}^j)\right\rangle + (\boldsymbol{\lambda}^*\mathbf{H})^* = 0, \\ &\frac{\partial f}{\partial\boldsymbol{\lambda}} = \mathbf{H}\boldsymbol{\theta} - \mathbf{g} = 0, \end{aligned} \tag{23}$$

the solution of which is given by

$$\boldsymbol{\lambda} = \left(\mathbf{H}\mathbf{A}^{-1}\mathbf{H}^*\right)^{-1}(\mathbf{H}\mathbf{A}^{-1}\mathbf{b} - \mathbf{g}), \tag{24a}$$

$$\boldsymbol{\theta} = \mathbf{A}^{-1}\left(\mathbf{b} - \mathbf{H}^*\boldsymbol{\lambda}\right), \tag{24b}$$

where

$$\mathbf{A} = \sum_j \left\langle (\mathbf{M}^j)^*\mathbf{R}^{-1}\mathbf{M}^j\right\rangle \quad \text{and} \quad \mathbf{b} = \sum_j \left\langle (\mathbf{M}^j)^*\mathbf{R}^{-1}(\mathbf{Z}^{j+1} - \mathbf{C}^j)\right\rangle. \tag{25}$$

4.2. Incorporating block decomposition for learning the high dimensional systems

As in most learning algorithms, the direct application of the algorithm developed in Section 3 can become inefficient for high dimensional systems with a large number of parameters. The low efficiency mainly comes from two aspects. First, in the E-Step, as the dimension of the model becomes large, the computational cost of solving the covariance matrix in the filter/smoother estimates may increase sharply since the dimension of the covariance matrix is the square of the dimension of the model. Second, in the M-Step, with the number of the parameters increases, the difficulty of searching the optimal parameters shoots up as well.

The block decomposition and divide and conquer methods [51,52,75] can be applied to many complex turbulent dynamical systems with multiscale structures [47], multilevel dynamics [48] or state-dependent parameterizations [49] that overcomes the two difficulties discussed above and leads to an extremely efficient learning scheme.

Let us start with a formal decomposition of the state variables

$$\mathbf{X} = \bigcup_{k=1}^{K} \mathbf{X}_k, \qquad \mathbf{Y} = \bigcup_{k=1}^{K} \mathbf{Y}_k \quad \text{with} \quad \mathbf{X}_k \in \mathbb{R}^{N_{\mathbf{X},k}} \text{ and } \mathbf{Y}_k \in \mathbb{R}^{N_{\mathbf{Y},k}},$$

where the total dimension of $\mathbf{X}$ and $\mathbf{Y}$ is $\mathrm{Dim}(\mathbf{X}) = \sum_{k=1}^{K} N_{\mathbf{X},k}$ and $\mathrm{Dim}(\mathbf{Y}) = \sum_{k=1}^{K} N_{\mathbf{Y},k}$, respectively. Correspondingly, the full dynamics in (1) are also decomposed into K groups, where the variables on the left hand side of the k-th group are $(\mathbf{X}_k, \mathbf{Y}_k)^{\mathrm{T}}$. In addition, for simplicity we assume both $\mathbf{B}_{\mathbf{X}}$ and $\mathbf{b}_{\mathbf{Y}}$ are diagonal and thus the noise coefficient matrices associated with the equations of $\mathbf{X}_k$ and $\mathbf{Y}_k$ are $\mathbf{B}_{\mathbf{X},k}$ and $\mathbf{b}_{\mathbf{Y},k}$, respectively.

4.2.1. Block decomposition of solving the conditional covariance (E-Step)

To apply the block decomposition strategy, the only requirement needs to be imposed on the dynamics of each $(\mathbf{X}_k, \mathbf{Y}_k)$ in (1) is as follows [51]. That is, the terms $\mathbf{A_0}_{,k}$ and $\mathbf{a_0}_{,k}$ can depend on all the components of $\mathbf{X}$ as in the original system while the terms $\mathbf{A_1}_{,k}$ and $\mathbf{a_1}_{,k}$ are only functions of $\mathbf{X}_k$, namely,

$$\begin{aligned} &\mathbf{A_0}_{,k} := \mathbf{A_0}_{,k}(t, \mathbf{X}), \quad \mathbf{a_0}_{,k} := \mathbf{a_0}_{,k}(t, \mathbf{X}), \\ &\mathbf{A_1}_{,k} := \mathbf{A_1}_{,k}(t, \mathbf{X}_k), \quad \mathbf{a_1}_{,k} := \mathbf{a_1}_{,k}(t, \mathbf{X}_k). \end{aligned} \tag{26}$$

In addition, only $\mathbf{Y}_k$ interacts with $\mathbf{A_1}_{,k}$ and $\mathbf{a_1}_{,k}$ on the right hand side of the dynamics of $\mathbf{X}_k$ and $\mathbf{Y}_k$. The initial values of $(\mathbf{X}_k, \mathbf{Y}_k)$ and $(\mathbf{X}_{k'}, \mathbf{Y}_{k'})$ for all $k' \neq k$ are also assumed to be independent with each other. Therefore, the equation of each $(\mathbf{X}_k, \mathbf{Y}_k)$ becomes

$$d\mathbf{X}_k = [\mathbf{A_0}(t, \mathbf{X}) + \mathbf{A_1}(t, \mathbf{X}_k)\mathbf{Y}_k]\, dt + \mathbf{B_X}(t, \mathbf{X}_k)\, d\mathbf{W_1}(t), \tag{27a}$$

$$d\mathbf{Y}_k = [\mathbf{a_0}(t, \mathbf{X}) + \mathbf{a_1}(t, \mathbf{X}_k)\mathbf{Y}_k]\, dt + \mathbf{b_Y}(t, \mathbf{X}_k)\, d\mathbf{W_2}(t). \tag{27b}$$

See [35,51,47,49,48] for many examples of such types of systems in geophysics, neuroscience and engineering. Note that in (27) each $(\mathbf{X}_k, \mathbf{Y}_k)$ is fully coupled with other $(\mathbf{X}_{k'}, \mathbf{Y}_{k'})$ for all $k' \neq k$ through $\mathbf{A_0}(t, \mathbf{X})$ and $\mathbf{a_0}(t, \mathbf{X})$. There is no trivial decoupling between different state variables.

Under such conditions, the evolution of the conditional covariance of $\mathbf{Y}_k$ (in the nonlinear filter (3)) conditioned on $\mathbf{X}$, namely $\mathbf{R_f}_{,k}$, has no interaction with that of $\mathbf{R_f}_{,k'}$ for $k' \neq k$ since $\mathbf{A_0}$ and $\mathbf{a_0}$ do not enter into the evolution of the conditional covariance. The evolution of different $\mathbf{R_f}_{,k}$ can thus be solved in a parallel way and the computation is extremely efficient due to the small size of each individual block [51]. So does the conditional covariance associated with the nonlinear smoother $\mathbf{R_s}_{,k}$. This greatly saves the computational cost in the E-Step.

4.2.2. Block decomposition of solving the likelihood function (M-Step)

Similar as the state variables, the parameters are also assumed to have a decomposition $\boldsymbol{\theta} = \bigcup_{k=1}^{K} \boldsymbol{\theta}_k$, where the parameters $\boldsymbol{\theta}_k$ appear only in the equations of $\mathbf{X}_k$ and $\mathbf{Y}_k$. Under such a condition, the likelihood function (14) can be further written as

$$\begin{aligned} \widetilde{\mathcal{L}} &= -J \log|\mathbf{R}^{-1}| + \mathrm{Tr}\left(\sum_j \left\langle \left(\mathbf{R}^{-1}(\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)(\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)^*\right)\right\rangle\right), \\ &= \sum_k \left(-J \log|\mathbf{R}_k^{-1}| + \mathrm{Tr}\left(\sum_j \left\langle \left(\mathbf{R}_k^{-1}(\mathbf{Z}_k^{j+1} - \mathbf{M}_k^j\boldsymbol{\theta}_k - \mathbf{C}_k^j)(\mathbf{Z}_k^{j+1} - \mathbf{M}_k^j\boldsymbol{\theta}_k - \mathbf{C}_k^j)^*\right)\right\rangle\right)\right). \end{aligned} \tag{28}$$

This allows a decomposition of the likelihood function. More specifically, based on the general formulae in (16) and (18), updating the covariance associated with the k-th block and the parameters $\boldsymbol{\theta}_k$ can be achieved via

$$\begin{aligned} \mathbf{R}_k &= \frac{1}{J}\sum_j \left\langle (\mathbf{Z}_k^{j+1} - \mathbf{M}_k^j\boldsymbol{\theta}_k - \mathbf{C}_k^j)(\mathbf{Z}_k^{j+1} - \mathbf{M}_k^j\boldsymbol{\theta}_k - \mathbf{C}_k^j)^* \right\rangle, \\ \boldsymbol{\theta}_k &= \left(\sum_j \left\langle (\mathbf{M}_k^j)^*\mathbf{R}_k^{-1}\mathbf{M}_k^j \right\rangle\right)^{-1} \left(\sum_j \left\langle (\mathbf{M}_k^j)^*\mathbf{R}_k^{-1}(\mathbf{Z}_k^{j+1} - \mathbf{C}_k^j) \right\rangle\right). \end{aligned} \tag{29}$$

Note that in many systems, different equations of $\mathbf{X}_k$ or $\mathbf{Y}_k$ (for $k = 1, \ldots, K'$) may share certain parameters. Such a situation can be easily adapted to the framework developed here. In fact, these shared parameters can be artificially treated as different parameters in the K' different groups. The learning algorithm is expected to provide nearly the same estimation values of these augmented parameters, assuming there is no observability issues. Then the averaged value can be used as the estimation of these shared parameters. An example will be shown in Section 6.2.

4.3. Sparse identification with LASSO

In practice, the exact model structure may not always be available. The starting model in the learning process typically involves a large number of components and parameters. Therefore, in addition to aiming for the accuracy of the resulting identified model, model parsimony also needs to be taken into account, which can prevent the potential overfitting issues. Some sparse model identification work can be found in [53–55,76,77]. Among different model parsimony criteria [78–80], the least absolute shrinkage and selection operator (LASSO) technique [56,57] is widely used and it can be incorporated into the learning framework developed here. With the LASSO regularization, the objective function in (13) is modified as

$$\min_{\boldsymbol{\theta},\mathbf{R}} \widetilde{\mathcal{L}}_{LASSO} = \min_{\boldsymbol{\theta},\mathbf{R}} \Big(\sum_j \Big\langle (\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)^* (\mathbf{R})^{-1} (\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j) \Big\rangle + J \log|\mathbf{R}| \Big) + \lambda \|\boldsymbol{\theta}\|_1, \tag{30}$$

where $\|\cdot\|_1$ is the l-1 norm and λ is the hyper parameter for the regularization. The advantage of the LASSO over ridge regression is that with a suitable choice of λ many of the parameters can be set to be nearly zero, which reaches the goal of model parsimony and facilitates the sparse identification of the model.

Despite that there is no closed form of the LASSO regression, an approximation optimization of the LASSO loss function can be achieved by [81]

$$\boldsymbol{\theta}^{(k+1)} = \left(\sum_j \Big\langle (\mathbf{M}^j)^* \mathbf{R}^{-1} \mathbf{M}^j \Big\rangle + \lambda \boldsymbol{\Psi}(\boldsymbol{\theta}^{(k)}) \right)^{-1} \left(\sum_j \Big\langle (\mathbf{M}^j)^* \mathbf{R}^{-1} (\mathbf{Z}^{j+1} - \mathbf{C}^j) \Big\rangle \right), \tag{31}$$

where $\Psi(\boldsymbol{\theta}^{(k)})$ is a diagonal matrix with the (i,i)-th entry being $1/|\theta_i^{(k)}|$, and $\boldsymbol{\theta}^{(k)} = (\theta_1^{(k)}, \theta_2^{(k)}, \ldots)^{\mathrm{T}}$. The superscript k stands for the k-th iteration of the parameters.

5. Quantifying the learning skill

One natural way of quantifying the learning skill is to compute the error in the recovered parameters related to the truth. However, it is hard to reach the conclusion that a small (large) error in the estimated model parameters corresponds to a high (low) skill in recovering the dynamical and statistical features of the underlying dynamical systems. In fact, in many complex turbulent dynamical systems, the dynamical and statistical features of the underlying dynamics can be very sensitive to the variation of certain parameters while they are relatively robust to the others. Therefore, some parameters are required to be learned with high accuracy while only a rough estimation of the others is sufficient. Another reason that prevents using the absolute or relative error in the recovered parameters as the quantification criterion is that the perfect model structure and parameter values are never known in practice. The only information available is the partially observed time series of the underlying systems.

In this article, the following procedure is used to assess the accuracy in the identified dynamics. First, run the model resulting from the learning algorithm forward. Then compare the model output of the observed variables with the available observations. It is nevertheless important to note that a point-to-point comparison between the model output trajectories and the observational time series is meaningless due to the chaotic and turbulent nature of the underlying dynamical systems. Thus, the following two quantities are adopted as the quantification criteria.

1. The *equilibrium PDF* represents the long-term statistical behavior of the system. Denote p and p^M the equilibrium PDF of the truth (observations) and that of the model output with the estimated parameters, a natural way to quantify the error in p^M compared with p is through the relative entropy $\mathcal{P}(p, p^M)$ [82–84],

$$\mathcal{P}(p, p^M) = \int p \log\left(\frac{p}{p^M}\right), \tag{32}$$

which is also known as the Kullback-Leibler divergence or information divergence [85–87]. Despite the lack of symmetry, the relative entropy has two attractive features. First, $\mathcal{P}(p, p^M) \geq 0$ with equality if and only if $p = q$. Second, $\mathcal{P}(p, p^M)$ is invariant under general nonlinear changes of variables. These provide an attract framework for assessing model errors in many applications [88–92,82,93,94].

2. The *temporal autocorrelation function (ACF)* is the correlation of a signal with a delayed copy of itself, as a function of delay. For a zero mean and stationary random process u, the autocorrelation function can be calculated as

$$R(t) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \frac{u(t+\tau)u^*(\tau)}{\mathrm{Var}(u)} d\tau, \tag{33}$$

where $\cdot^*$ denotes the complex conjugate. Different from the equilibrium PDF, the autocorrelation function characterizes the memory of the system and involves the fundamental dynamical feature. The simplest way to compute the error in the ACF associated with the identified model is to compute its difference related to that of the true signal, and a more systematic way is to adopt a spectral information criteria, the details of which can be found in [95].

From the discussions above, it is clear that if the error in both the PDF and ACF is small, then the model resulting from the learning algorithm is at least able to capture the stationary statistics and the leading order dynamical characteristics of the underlying system. Note that these two criteria are simply the necessary conditions for reaching the conclusion that the resulting model from the learning algorithm is skillful in recovering the key features of the true dynamics.

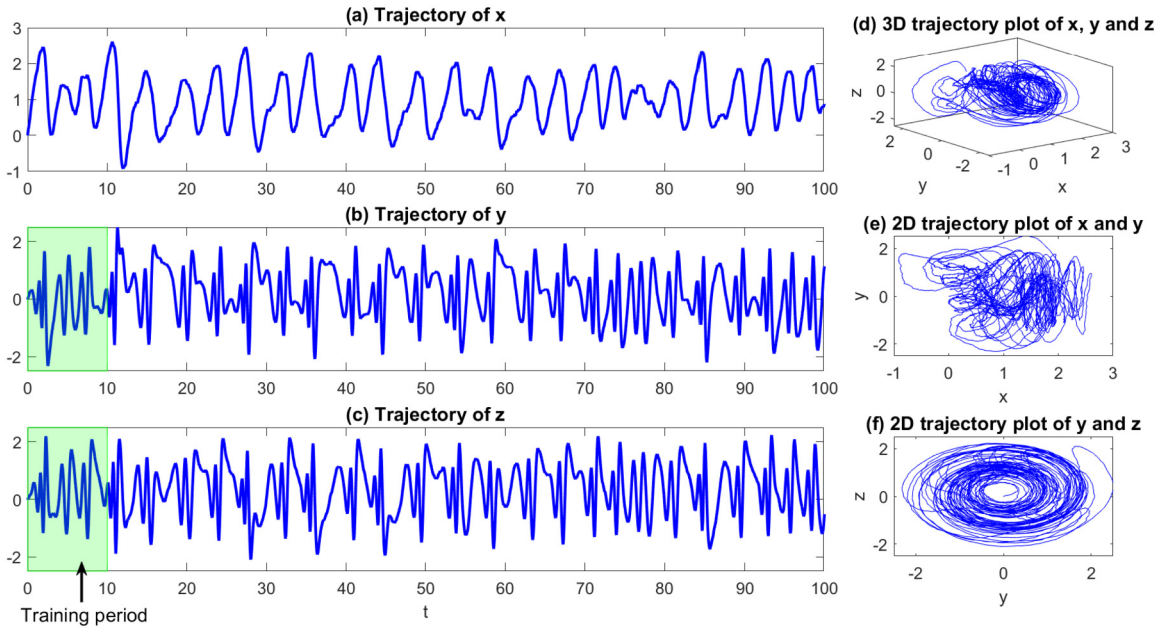

Fig. 1. Trajectories of the L84 model (34). Panels (a)–(c): 1D trajectory of x, y and z respectively. Panel (d): 3D trajectory plot of x, y and z. Panel (e): 2D trajectory plot of x and y. Panel (f): 2D trajectory plot of y and z. The green shading areas in Panels (b)–(c) indicate the short training period. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

6. Test examples of learning parameters in nonlinear and non-Gaussian models using short training data

6.1. A perfect model test on a simple analogue of the global atmospheric circulation model

We start with the following low-order atmospheric general circulation model, which was introduced by Lorenz [96,97]. It is sometimes named as the Lorenz 1984 (or L84) model. The model reads

$$\begin{aligned} dx &= \big(-(y^2+z^2) - ax + f\big)dt + \sigma_x dW_x, \\ dy &= \big(-bxz + xy - y + g\big)dt + \sigma_y dW_y, \\ dz &= \big(bxy + xz - z\big)dt + \sigma_z dW_z. \end{aligned} \tag{34}$$

In (34), the zonal flow x represents the intensity of the mid-latitude westerly wind current (or the zonally averaged meridional temperature gradient, according to thermal wind balance), and a wave component exists with y and z representing the cosine and sine phases of a chain of large-scale vortices superimposed on the zonal flow. Relative to the zonal flow, the wave variables are scaled so that $x^2+y^2+z^2$ is the total scaled energy (kinetic plus potential plus internal). These equations can be derived as a Galerkin truncation of the two-layer quasigeostrophic potential vorticity equations in a channel. Note that in the original version of the L84 model [96], the forcing in the first equation of (34) was written as $a\tilde{f}$. In (34), f is defined as $f = a\tilde{f}$, where estimating a and $\tilde{f}$ is equivalent to estimating a and f.

The following parameters are used to generate the true signal in the test here [35],

$$g = 1, \qquad b = 4, \qquad a = 1/4, \qquad f = 2, \qquad \text{and} \qquad \sigma_x = \sigma_y = \sigma_z = 0.1. \tag{35}$$

A model simulation is illustrated in Fig. 1. Panels (b)–(c) show the trajectories of the large-scale vorticity components y and z, which have intermittent oscillation structures with chaotic behavior. According to the phase plot Panel (f), the components y and z clearly form a pair of oscillator. The wind current component x (Panel (a)) is also quasi-periodic and it intermittently goes easterly when the sign becomes negative. The variable x is highly correlated with the large-scale vorticity components (Panels (d)–(e)).

Now assume the structure of the L84 model is known and the two large-scale vorticity components y and z are observed. The training period for learning the model parameters is indicated by the green shading area in Panels (b)–(c) of Fig. 1. This short training period only includes three quasi-periods of oscillations and it also contains a transient phase at the very beginning. The initial guesses of the parameters in the learning process are all assumed to be doubled compared with the true values in (35). The trace plots of the estimated parameters are shown in Fig. 2. After 50 iterations, the parameters converge to nearly the true values.

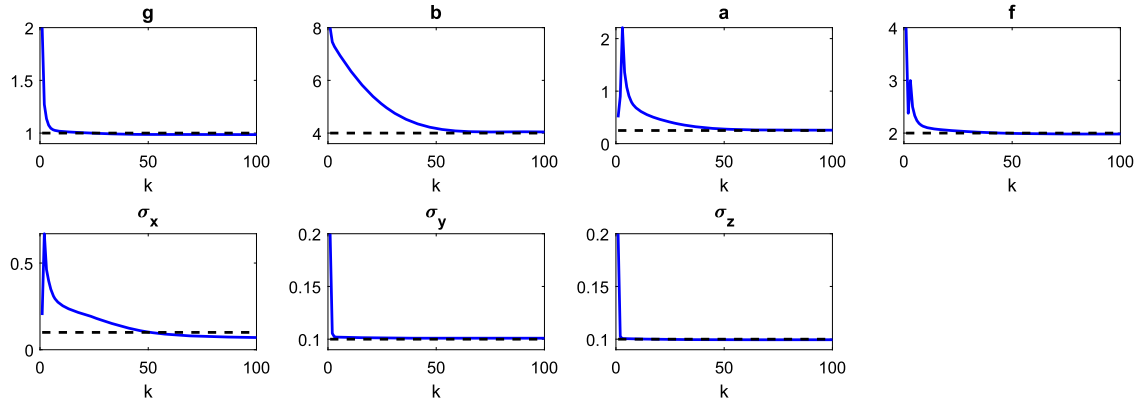

Fig. 2. Trace plot. Learning the parameter of the L84 model (34) in a perfect model setting. The true parameter values are shown in black dashed lines.

To test the sensitivity of the learning algorithm, different random number seeds and different initial values being randomly drawn from the equilibrium statistics have been used to generate the short training period. The same parameter estimation results are obtained, which implies the robustness of the algorithm.

6.2. A spatial-extended stochastically coupled FitzHugh-Nagumo (FHN) model for excitable medium

The FitzHugh-Nagumo model (FHN) is a prototype of an excitable system, which describes the activation and deactivation dynamics of a spiking neuron [39]. The stochastic versions of the FHN model have been widely studied and applied in the context of stochastic resonance, which processes the features of the noise-induced limit cycles [98–103]. Here, a spatially-extended stochastically coupled FHN model is used as a test model, which has attracted much attention in practice as a noisy excitable medium [104–107],

$$\begin{aligned} \epsilon du_i &= \left(\tilde{d}_u(u_{i+1}+u_{i-1}-2u_i)+u_i-\frac{1}{3}u_i^3-v_i\right)dt+\sqrt{\epsilon}\tilde{\delta}_u dW_{u_i}, \\ dv_i &= \left(u_i+\tilde{a}\right)dt+\tilde{\delta}_v dW_{v_i}, \qquad i=1,\ldots,N. \end{aligned} \tag{36}$$

In (36), both u_i and v_i are functions of space with index i, where u_i can be regarded as the membrane potential and different v_i then represent the recovery variables. The factor ϵ represents a time scale ratio, which is much smaller than one ($\epsilon \approx 10^{-2}$), implying that $u_i(t)$ are the fast and $v_i(t)$ are the slow variables. The parameter $\tilde{a}$ plays a key role for bifurcation in the deterministic case, where the system has a fixed point for $\tilde{a} > 1$ and has a limit cycle otherwise. Different u_i and v_i are coupled via the diffusion term with strength $\tilde{d}_u$.

For the convenience of statement, we slightly change the notation in system (36),

$$\begin{aligned} du_i &= \left(d_{u,i}(u_{i+1}+u_{i-1}-2u_i)+e_i(u_i-\frac{1}{3}u_i^3-v_i)\right)dt+\sigma_{u,i}dW_{u_i}, \\ dv_i &= \left(u_i+a_i\right)dt+\sigma_{v,i}dW_{v_i}, \qquad i=1,\ldots,N. \end{aligned} \tag{37}$$

The correspondence between the parameters in (37) and (36) can be easily established. Here we consider an inhomogeneous model and allow some of the parameters $(e_i, d_{u,i}, a_i, \sigma_{u,i}, \sigma_{v,i})$ to be functions of the space location i. The total number of the spatial grid points is set to be $N = 25$. To generate the true signal, the following parameters are used,

$$\begin{aligned} e_i &= 100+30\cos(2\pi i/N), \qquad a_i = 1+0.5\cos(2\pi i/N), \qquad d_{u,i}=3, \\ \sigma_{u,i} &= 1+0.5\sin(2\pi i/N), \qquad \sigma_{v,i}=0.8, \end{aligned} \tag{38}$$

where e_i, a_i and σ_{u_i} vary in space while $d_{u,i}$ and $\sigma_{v,i}$ are constants. Note that the magnitudes of the true parameters in (38) are very different, which leads to a tough test of learning this spatial-extended stochastically coupled FHN model. Below, assume the observational variables are u_i. Since there is no prior knowledge on the parameters, all the parameters are assumed to be functions of i in the learning process, as was discussed in Section 4.2. Therefore, the total number of the parameters to be determined in the learning algorithm is $5N = 125$ and the total dimension of the coupled model (37) is $2N = 50$. One important feature of the stochastically coupled FHN model (37) is that it satisfies the conditions described in Section 4.2. Thus, the block decomposition technique can be applied for learning the parameters in this high-dimensional system. Finally, as a remark, the choice of these parameters are consistent with those in [51,108] for a weakly coherent spatial structure regime.

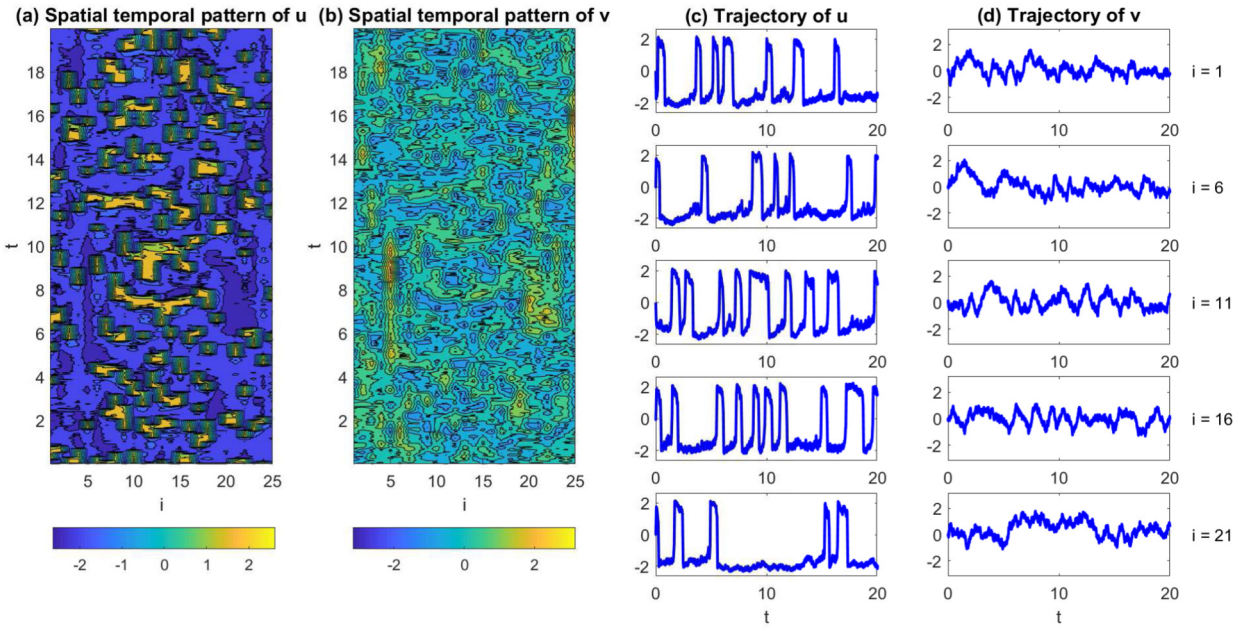

Fig. 3. Panels (a)–(b): A realization of the stochastically coupled FHN model (37). Panels (c)–(d): The associated time series at different locations $i = 1, 6, 11, 16$ and 21. The spatial temporal pattern shown in Panel (a) is used as the training data.

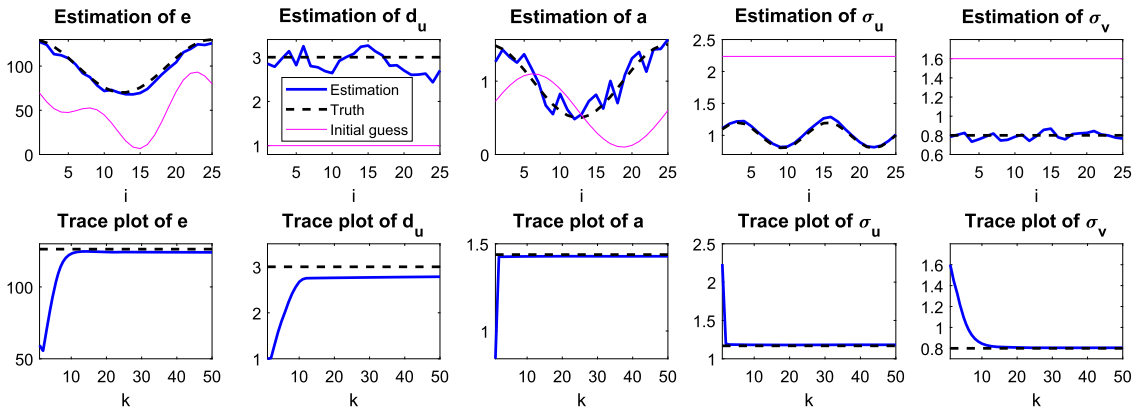

Fig. 4. Learning the parameters of the stochastically coupled FHN model (37). The panels in the top row show the true values (black dashed curves), the initial guess of the parameters (pink thin curves) and the estimated values after 50 iterations (blue curves). The panels in the bottom row show the trace plots at $i = 2$.

Panels (a)–(c) of Fig. 3 illustrate a realization of the model (37) with the true parameters (38), where the weakly coherent spatial structure is easily seen. The associated time series at fixed spatial locations are shown in Panels (c)–(d). The time series of u are all intermittent with extreme events, the associated PDF of which is clearly non-Gaussian. The frequency of the extreme events occurrence is quite distinct at different locations due to the fact that a spatial varying function of a_i is used here. The short period with only 20 units as shown in Panel (a) is used as the training data for learning the model parameters of the stochastically coupled FHN model (37).

The parameter learning results are illustrated in Fig. 4. The panels in the top row show the true values (black dashed curves), the initial guesses of the parameters (pink thin curves) and the estimated values after 50 iterations (blue curves). The panels in the bottom row show the trace plots at $i = 2$. The trace plots at other spatial locations have similar behavior. From these plots, it is clear that despite a large error in the initial guesses, the algorithm provides a quite accurate estimation of the parameters within a few iterations. Although there are some small errors in the estimated parameter d_u, the model simulation based on the estimated parameters and the associated ACFs and PDFs nevertheless highly resemble those using the perfect parameter values including reproducing the extreme burst events in u_i while that using the initial guesses of the parameters are completely different. The reason that the estimated d_u does not perfectly match the truth is that the model simulation and the associated statistical features are quite robust with respect to d_u around its true value. The short training period and the noise in the model also account for such a small bias.

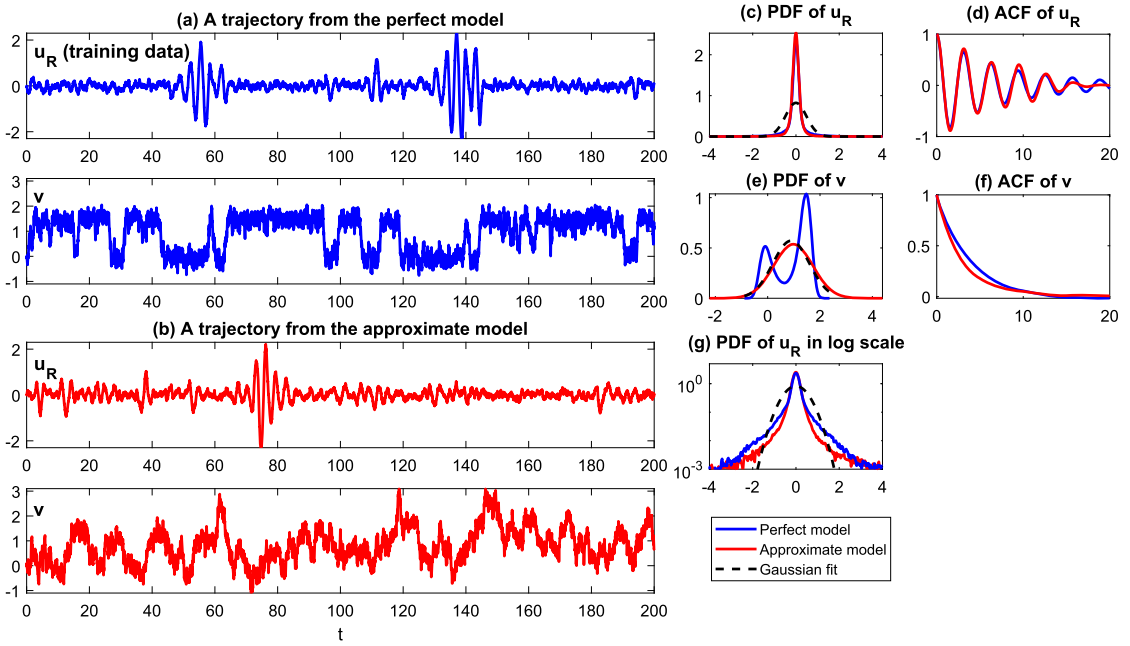

Fig. 5. Panel (a): A realization from the perfect model (39), where the signal of u_R here (and u_I) with 200 units is used as the training period. Panel (b): A realization from the approximate model (42) with the estimated parameters. Panels (c) and (d): Comparison of the PDFs and ACFs of u_R. Panels (e) and (f): Comparison of the PDFs and ACFs of v. The comparison of the PDFs of u_R in logarithm scale is shown in Panel (g). The dashed curve in Panel (e) shows the Gaussian fit of the non-Gaussian PDF of v associated with the perfect model. The ACFs and the PDFs are computed based on long trajectories with 5000 units.

6.3. Learning model parameters with model error

In this subsection, we aim at studying learning the parameters in the presence of model error. Consider the following coupled nonlinear system as the perfect model that generates the true signal,

$$\begin{aligned} \mathrm{d}u_R &= (-vu_R - \omega u_I + f_{uR})\,\mathrm{d}t + \sigma_{uR}\,\mathrm{d}W_{uR}, \\ \mathrm{d}u_I &= (-vu_I + \omega u_R + f_{uI})\,\mathrm{d}t + \sigma_{uI}\,\mathrm{d}W_{uI}, \\ \mathrm{d}v &= (-a_v v + b_v v^2 - c_v v^3 + f_v)\,\mathrm{d}t + \sigma_v\,\mathrm{d}W_v. \end{aligned} \tag{39}$$

If we define $u = u_R + iu_I$, then the first two equations can be combined as

$$\mathrm{d}u = \big((-v + i\omega)u + f_u\big)\,\mathrm{d}t + \sigma_u\,\mathrm{d}W_u. \tag{40}$$

In (39) or (40), the variable v acts as a stochastic damping of u. The parameter ω represents the oscillation frequency while f_u stands for deterministic forcing. The process of v contains a cubic nonlinearity, which allows two stable fixed points in the deterministic setup. Therefore, in the presence of the random noise, the statistics of v can be non-Gaussian with a bimodal distribution. Such a toy model can be used as a crude approximation to describe some large-scale phenomena with oscillation structures and intermittency in nature, such as the MJO and the monsoon [109]. The following parameters are adopted to generate the true signal,

$$\begin{aligned} &\omega = 2, \qquad f_{uR} = f_{uI} = 0, \qquad \sigma_{uR} = \sigma_{uI} = 0.2, \\ &a_v = 2.8, \qquad b_v = 8, \qquad c = 4, \qquad f_v = -0.4, \qquad \sigma_v = 0.7. \end{aligned} \tag{41}$$

These parameters allow the trajectory of v to be nearly two-state, where one state has positive value that stabilizes the signal of u and the other one is slightly negative that leads to an intermittent amplification of the signal of u. A realization of the perfect model is shown in Panel (a) of Fig. 5, where the signal of u is used for training the model. Within this 200-unit period, there are only two major intermittent events and a few weak ones. It is thus clear that the training data is very limited.

In many practical situations, stochastic parameterizations are used to simplify certain complicated dynamical components that allow efficient computations [110–112]. Here, a simple stochastic parameterization is used for the process v and the resulting approximate model reads,

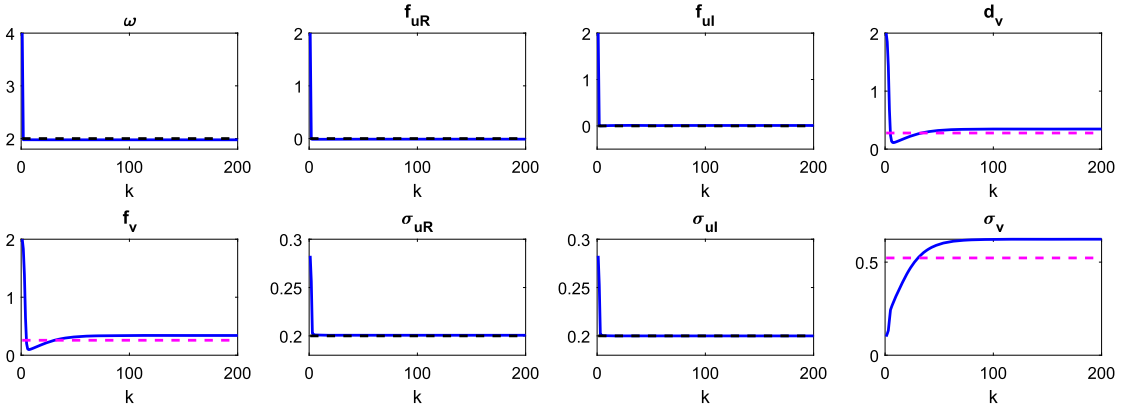

Fig. 6. Trace plots of the parameter estimation of the approximate model (42). The true signal is generated from (39) and the signal of u is used as the input of estimating the parameters in the approximate model (42). The black dashed lines indicate the true values in the perfect model while the pink dashed lines indicates the parameters corresponding to the Gaussian statistics of v in the perfect model (39).

$$\begin{aligned} \mathrm{d}u_R &= (-vu_R - \omega u_I + f_{uR})\,\mathrm{d}t + \sigma_{uR}\,\mathrm{d}W_{uR}, \\ \mathrm{d}u_I &= (-vu_I + \omega u_R + f_{uI})\,\mathrm{d}t + \sigma_{uI}\,\mathrm{d}W_{uI}, \\ \mathrm{d}v &= (-d_v v + f_v)\,\mathrm{d}t + \sigma_v\,\mathrm{d}W_v. \end{aligned} \tag{42}$$

In other words, the nonlinear process of v in (39) is replaced by a linear Gaussian process in (42). Nevertheless, the coupled system in (42) remains nonlinear and the stochastic damping continues to play a key role in generating the intermittency in the signal of u. Since v is regarded as a process that is stochastically parameterized, it is natural to assume that only the large-scale variable u is observed. The goal here is to learn the parameters in the approximate model (42) given the partially observed true signal of only the variable u generated from the perfect system (39). Notably, by observing u, the approximate model (42) is a conditional Gaussian nonlinear model while the original system (39) is not.

The trace plots of the estimated parameters in the approximate model (42) are shown in Fig. 6. It is clear that the parameters associated with the u process almost perfectly match the true values in (41) (black dashed lines). For the three parameters d_v, f_v and σ_v in the stochastic parameterized process of v, there are no ground truth to compare. Nevertheless, these estimated parameters almost match the values corresponding to the Gaussian statistics of v in the perfect model (39) (pink dashed lines). In Panel (b) of Fig. 5, a realization of the approximate model with the estimated parameters is shown. The path-wise behavior of u looks similar to the truth with an intermittent appearance of extreme events. On the other hand, the trajectory of the hidden variable v is different from the truth, where in the approximate model, there is no obvious two-state behavior due to its linear and Gaussian dynamics. Nevertheless, the time series of v in Panel (b) intermittently goes below the threshold $v = 0$, which leads to the intermittent instability in the time series of u. Panels (c)–(f) compare the PDFs and ACFs of u and v using the perfect and approximate models. The approximate model is able to generate similar fat-tailed non-Gaussian statistics of the observed variable u with only a slight underestimation of the tails. The fat-tailed PDF in the approximate model indicates its skill in reproducing the observed extreme events as in the perfect model. The approximate model is also able to generate a Gaussian PDF that almost overlaps with the Gaussian fit of the bimodal distribution of v as in the perfect model.

6.4. Role of the physics constraint

In all the examples shown above, physics constraint was already incorporated in the model structure. Yet, the physics constraint cannot always be identified in the learning process, especially when both strong nonlinear interactions and noise appear in the unobserved process. In such a situation, incorporating the physics constraint becomes crucial in the learning process to prevent the pathological behavior in the recovered dynamics.

Consider the following nonlinear triad model where u is the observed variable while the nonlinearity appears only in the unobserved variables A and B,

$$\begin{aligned} \mathrm{d}u &= (-d_u u + c_1 B)\,\mathrm{d}t + \sigma_u\,\mathrm{d}W_u, \\ \mathrm{d}A &= (-d_A A - \gamma_1 B u)\,\mathrm{d}t + \sigma_A\,\mathrm{d}W_A, \\ \mathrm{d}B &= (-d_B B + \gamma_2 A u + c_2 u)\,\mathrm{d}t + \sigma_B\,\mathrm{d}W_B. \end{aligned} \tag{43}$$

This model is a low-order version of the Charney-DeVore flows [113,114], describing a barotropic zonally unbounded flow over a sinusoidal topography in a zonal channel with quasigeostrophic dynamics. In (43), u is the zonal mean flow while A

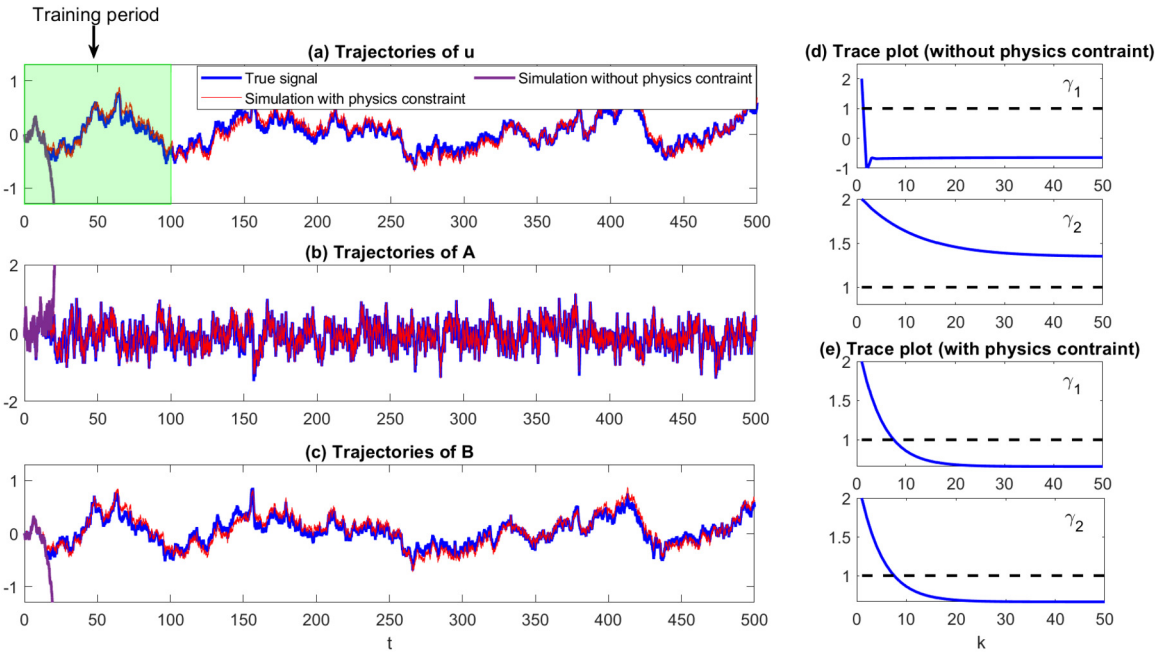

Fig. 7. Panels (a)–(c): model simulation of the coupled low-order CDV model (43). The blue curves show the simulation with the true parameters (44). The purple curves show the simulations with the estimated parameters without the physics constraint (Panel (d)) while the red curves show those with physics constraints (Panel (e)). The same random number seeds are used in these three simulations. The short true signal of u in the green shading area stands for the training period. Panels (d) and (e) are the trace plots of the estimated parameters, where physics constraint is not and is included in the algorithm.

and B are the sine and cosine amplitudes of a wave component, which are the leading order terms resulting from a truncation expansion [113]. The only nonlinearity comes from the terms $-\gamma_1 Bu$ and $\gamma_2 Au$, both appearing in the unobserved processes.

The following parameters are used to generate the true signal in the test here,

$$\sigma_u = \sigma_B = 0.1, \qquad \sigma_A = 0.5, \qquad \text{All others } = 1. \tag{44}$$

Note that these parameters are not the most relevant ones to the Charney-DeVore flows but these parameter values help illustrate the importance of including the physics constraint into the learning algorithm. In Panels (a)–(c) of Fig. 7, the blue curves show a realization of the perfect model, where the short period of the true trajectory of u within the green shading area stands for the training data. Note that the trajectory of A is more noisy than the other two components while A is only directly linked with the process of another unobserved variable B and it does not appear in the observed process.

Below, to reduce the degree of freedom in the parameter searching space and minimize the interference from the linear part of the model, we assume all the other parameters are perfectly known and the only parameters that need to be learned are the ones related to the nonlinear terms, namely γ_1 and γ_2. The initial guesses of these two parameters are both doubled from the perfect values.

Panels (d) and (e) show the trace plots of the learned parameters γ_1 and γ_2. The algorithm corresponding to the result in Panel (d) does not take into account the physics constraint while the physics constraint $\gamma_1 = \gamma_2$ is incorporated in the algorithm associated with Panel (e). Clearly, without the physics constraint, the estimated parameters γ_1 and γ_2 converge to complete different values. What is worse, equipped with these estimated parameters, a free run of the coupled model (43) quickly leads to a blow-up solution. See the purple curves in Panels (a)–(c). This is not surprising since the learning algorithm is based on the likelihood function, which does not necessity guarantee the long-term properties of the system, especially in the presence of large noise and partial observations. On the other hand, with the physics constraint, the estimated parameters of γ_1 and γ_2 converge to the same value. Despite some difference from the truth, the model simulation (and the associated ACFs and PDFs) based on these estimated parameters highly resembles the truth. See the red and blue curves in Panels (a)–(c), where the random number seeds are set to be the same in these simulations. The error in the estimated parameters is due to the large noise in the process of A and the weak and indirect feedback from A to the observed variable u. In other words, the response of the observed variable u to the variation of these parameters is quite robust around the perfect values. This also indicates that computing the absolute error in the estimated parameters is not always a good choice in assessing the learning skill, which was discussed in Section 5.

Table 1
Comparison of the parameters in L84 model.

(a) Perfect model (L84)									
	L_{ij}				B_{ij}			F_j	σ_j
0.25	0.00	0.00	-1.00	-1.00	0.00	0.00	0.00	2.00	0.10
0.00	-1.00	0.00	0.00	0.00	0.00	1.00	-4.00	1.00	0.10
0.00	0.00	-1.00	0.00	0.00	0.00	4.00	1.00	0.00	0.10
(b) Identified model									
	L_{ij}				B_{ij}			F_j	σ_j
-0.30	0.09	-0.01	-1.54	-1.55	0.00	-0.35	0.23	3.12	0.34
-0.01	-1.09	0.59	0.11	-0.28	-0.20	0.64	-2.66	0.94	0.10
0.02	-0.64	-1.18	0.31	0.08	0.35	2.68	0.64	-0.09	0.10
(c) Identified model with LASSO									
	L_{ij}				B_{ij}			F_j	σ_j
-0.14	0.00	0.00	-1.58	-1.61	0.00	0.00	0.00	3.00	0.34
0.00	-0.89	0.00	0.00	0.00	0.00	0.54	-2.31	0.88	0.10
0.00	0.00	-0.93	0.00	0.00	0.00	2.32	0.55	0.00	0.10

7. Model identification with unknown model structure

In all the previous test examples, the model ansatz was given. Yet, in some applications, the exact form of the model is not available before hand. Both the model structure and parameters need to be learned from the noisy partial observations. The goal in this section is to study the skill of learning the nonlinear dynamics without known the exact model structure and investigate whether the dynamics of the observed and unobserved variables can always be learned or identified from the partially observed data.

The tests here focus on the situations in which the observation is given by a pair of time series, denoted by (y, z). To allow a sufficient large degree of freedom in the learning process, an extra hidden process x is added to the starting model. In other words, a three dimensional system with two components representing the observed variables is adopted here for learning the unknown dynamics. As in many learning algorithms [53–55,76,77], we construct a library consisting of candidate nonlinear functions to this model. The difference in the setup here is that partial observations and random noise appear in the learning process. As a simple illustration, assuming the highest order nonlinearity being quadratic, then the following general triad system is used as the starting model for the learning process,

$$\begin{aligned}
\mathrm{d}x &= (L_{11}x + L_{12}y + L_{13}z \\
&\quad + B_{11}y^2 + B_{12}z^2 + B_{13}yz + B_{14}xy + B_{15}xz + F_1)\,\mathrm{d}t + \sigma_1\,\mathrm{d}W_1, \\
\mathrm{d}y &= (L_{21}x + L_{22}y + L_{23}z \\
&\quad + B_{21}y^2 + B_{22}z^2 + B_{23}yz + B_{24}xy + B_{25}xz + F_2)\,\mathrm{d}t + \sigma_2\,\mathrm{d}W_2, \\
\mathrm{d}z &= (L_{31}x + L_{32}y + L_{33}z \\
&\quad + B_{31}y^2 + B_{32}z^2 + B_{33}yz + B_{34}xy + B_{35}xz + F_3)\,\mathrm{d}t + \sigma_3\,\mathrm{d}W_3.
\end{aligned} \tag{45}$$

In (45), the terms with L_{ij} are the linear terms while those with B_{ij} are the quadratic terms. The three F_i terms are deterministic forcing and the three σ_i coefficients are stochastic forcing. Note that in order for the model to fit into the conditional Gaussian framework, there is no x^2 terms appear on the right hand side of (45).

7.1. The Lorenz 84 model

The first test here is the noisy L84 model (34) as was introduced in Section 6.1. The true parameters corresponding to the general form (45) are listed in the top part of Table 1, which are the same as those in (35). The initial guesses of the parameters are $L_{11} = L_{22} = L_{33} = -1$, $B_{25} = -2$, $B_{34} = 2$, $\sigma_x = \sigma_y = \sigma_z = 0.5$ with all other parameters being zero. The model with these parameters simply gives a damped regular nonlinear oscillator of (y, z), which is very far from the true dynamics of the L84 model. The learning algorithm is run for 25 iterations, at the end of which the curves of the learned parameters converge.

7.1.1. Learning results

The structure and parameters resulting from the learning algorithm are listed in Part (b) of Table 1, which however seem to be very different from the perfect model, namely the L84 model. Surprisingly, both the path-wise and statistical features of the observed variables (y, z) associated with the identified and the perfect model are nearly identical to each other, including the bimodal non-Gaussian PDFs and the significant non-exponential decaying ACFs. See Panels (b)–(c), (e)–(f) and (h)–(i) of Fig. 8. These results actually indicate the success of nearly perfectly learning the dynamics of the two observed variables (y, z).

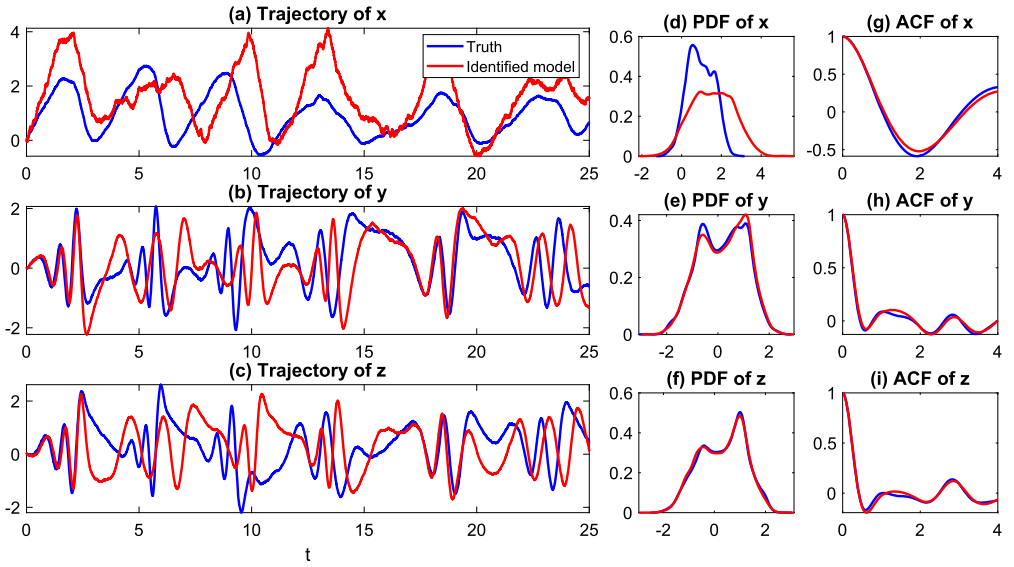

Fig. 8. Comparison of the trajectories (Panels (a)–(c)), the PDFs (Panels (d)–(f)) and the ACFs (Panels (g)–(i)) of the true (blue) and identified model (red), where the true model is the L84 model. The trajectories of both the true and the identified models are simply from one random realization and they are not expected to have one-to-one point-wise correspondence. The true trajectories in Panels (b)–(c) are used as the training data. The ACFs and the PDFs are computed based on long trajectories with 1000 units.

On the other hand, the error in the recovered trajectory and statistics of the unobserved variable x in the identified model is non-negligible compared with the truth. More specifically, the amplitude of x in the identified model is overestimated and the associated PDF is more symmetric. Nevertheless, since the true signal of x is unknown, quantifying the accuracy of the identified model can only be based on the recovered features of the observed variables, as was discussed in Section 5. Therefore, the identified model succeeds in learning the observed nonlinear and chaotic features of the true dynamics.

Now we focus on the coefficients learned in the identified model (Part (b) in Table 1). Compared with the true parameters, there are three main differences. First, the key parameters B_{25} and B_{34} are underestimated. These two parameters appear in front of the terms xz and xy in the y and z equations, respectively, and play the role of triggering irregular nonlinear oscillations. As a compensation of the underestimation of B_{25} and B_{34}, the amplitude of x is strengthened due to the combination of the enhanced positive feedback forcing term F_1 and negative nonlinear feedback coefficients B_{11} and B_{12}. On the other hand, the two coefficients that correspond to the linear oscillations L_{23} and L_{32} become significantly nonzero as well. The overall contributions from these three major changes retain the same chaotic features in the observed variables y and z as in the perfect model.

7.1.2. Overfitting and model predictability

Noticing that the identified model has a more complicated structure than the perfect model, a natural question to ask is whether the identified model suffers from overfitting.

To answer this question, let us first focus on the model prediction skill. Despite sharing the same underlying principles, it is important to note the difference between the approach used here and the state-of-art supervised learning methods. In fact, incorporating the statistics of the unobserved variables into the likelihood function helps alleviate noise in the output value, which to some extent reduces the risk of overfitting. Another important fact is that instead of predicting a single path, the ensemble/statistical approach is often adopted for forecasting complex nonlinear turbulent dynamical systems. In other words, the prediction itself takes into account the uncertainty, which is different from the deterministic forecast or the classification problems. Notably, the results in Panels (e) and (f) of Fig. 8 show that the identified model is able to predict the long range non-Gaussian statistics of the perfect system. In addition, the similarity in the ACFs in Panels (h) and (i) indicates that the internal prediction skill or the predictability of (y, z) of the identified model is almost the same as that of the perfect model. The identical ACF of the hidden variable x as shown in Panel (g) also illustrates that the identified model and the perfect model share the same feedback mechanism from x to (y, z). These facts lead to the conclusion that the two models have the same statistical forecast skill of the observed variables (y, z).

On the other hand, from the aspect of model parsimony, there is still a room for the improvement of the identified model. A simpler model is often preferred because when the model structure becomes complicated the model response may be sensitive to the perturbation of some parameters. Notably, capturing the model sensitivity is equally important as recovering the model fidelity (the equilibrium statistics). This is because the model sensitivity plays an important role in predicting the model response to the variation of certain external perturbations, which appear in many real applications including the situation of climate change [115,116]. In practice, some extra constraints from physics, observations or experiments often

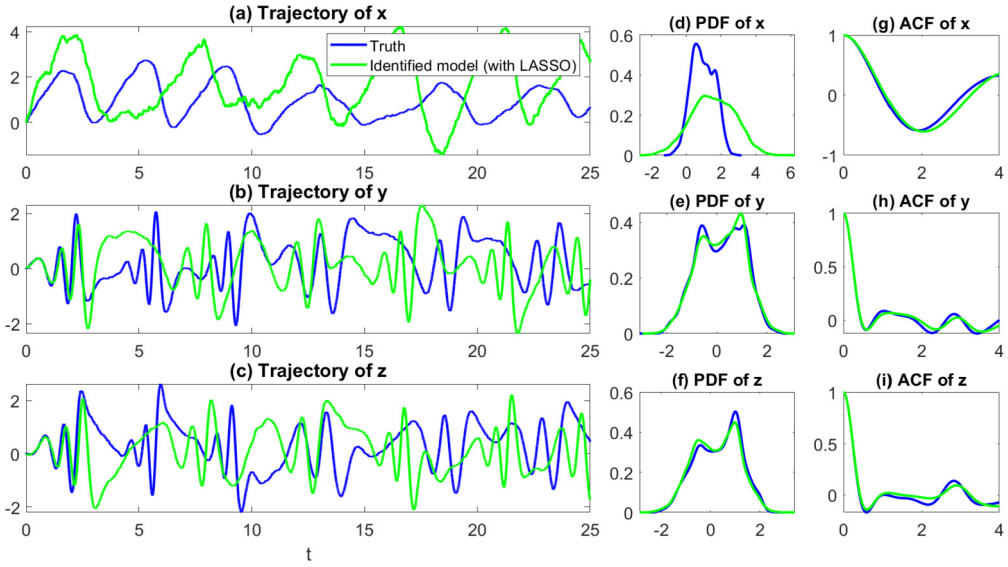

Fig. 9. Comparison of the trajectories (Panels (a)–(c)), the PDFs (Panels (d)–(f)) and the ACFs (Panels (g)–(i)) of the true (blue) and identified model (red), where the true model is the L84 model. The trajectories of both the true and the identified models are simply from one random realization and they are not expected to have one-to-one point-wise correspondence. The true trajectories in Panels (b)–(c) are used as the training data. The LASSO is applied here to find the sparse solution. The ACFs and the PDFs are computed based on long trajectories with 1000 units.

contribute to the development of a parsimonious model. Otherwise, the sparse identification technique can be adopted for reducing the model complexity.

7.1.3. Sparse model identification

To reduce the model complexity, the LASSO technique is incorporated into the model identification framework. By choosing the penalty hyper parameter $\lambda = 0.001$ in (30), the identified model is shown in Part (c) of Table 1, where the parameters with amplitude smaller than 0.01 have been set to be zero. Comparing these parameters with those in the perfect model, it is clear that the identified model has the same model complexity as the perfect one. However, the parameters in the identified model are yet quite different from the truth. Fig. 9 includes a comparison of the model simulations and statistics between the perfect model (blue) and the identified one with LASSO (green). Again, it is found that both the path-wise and statistical features of the observed variables y and z remain very similar to the truth. This indicates that the model is not uniquely identified in the presence of partial observations and noise even if the complexity of the identified model is the same as the perfect one. As a final test, if the model structure is given and only the parameters of the L84 model are required to be estimated, then starting from the initial values used here, the model parameters will converge to the true values.

7.2. The noisy Lorenz 63 model

As a second test, consider the following Lorenz 63 (L63) model [117],

$$dx = \sigma(y - x)dt + \sigma_x dW_x, \tag{46a}$$

$$dy = \big(x(\rho - z) - y\big)dt + \sigma_y dW_y, \tag{46b}$$

$$dz = (xy - \beta z)dt + \sigma_z dW_z, \tag{46c}$$

the deterministic version of which was proposed by Lorenz in 1963 [117]. It is a simplified mathematical model for atmospheric convection. The equations relate the properties of a two-dimensional fluid layer uniformly warmed from below and cooled from above. In particular, the equations describe the rate of change of three quantities with respect to time: x is proportional to the rate of convection, y to the horizontal temperature variation, and z to the vertical temperature variation. The constants σ, ρ, and β are system parameters proportional to the Prandtl number, Rayleigh number, and certain physical dimensions of the layer itself [118]. The L63 model is also widely used as simplified models for lasers, dynamos, thermosyphons, electric circuits, chemical reactions and forward osmosis [119–125]. The noisy version of the L63 model includes more turbulent and small-scale features and their interactions with the three large scale variables while it retains the characteristics in the original L63.

In order to test the parameter estimation skill, the following parameters are used to generate the true signal,

$$\sigma = 10, \qquad \rho = 28, \qquad \beta = 8/3, \qquad \sigma_x = \sigma_y = \sigma_z = 2. \tag{47}$$

Table 2
Comparison of the parameters in L63 model.

Perfect model (L63)									
	L_{ij}				B_{ij}			F_j	σ_j
-10.00	10.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00
28.00	-1.00	0.00	0.00	0.00	0.00	0.00	-1.00	0.00	2.00
0.00	0.00	-2.67	0.00	0.00	0.00	1.00	0.00	0.00	2.00
Identified model									
	L_{ij}				B_{ij}			F_j	σ_j
-23.15	0.56	1.12	0.00	-0.01	0.04	0.25	0.42	-29.47	0.89
107.35	10.89	-6.90	0.00	0.08	-0.42	0.00	-3.92	127.91	2.00
-0.01	5.46	-2.80	0.41	0.00	-0.11	4.16	0.00	1.81	2.00

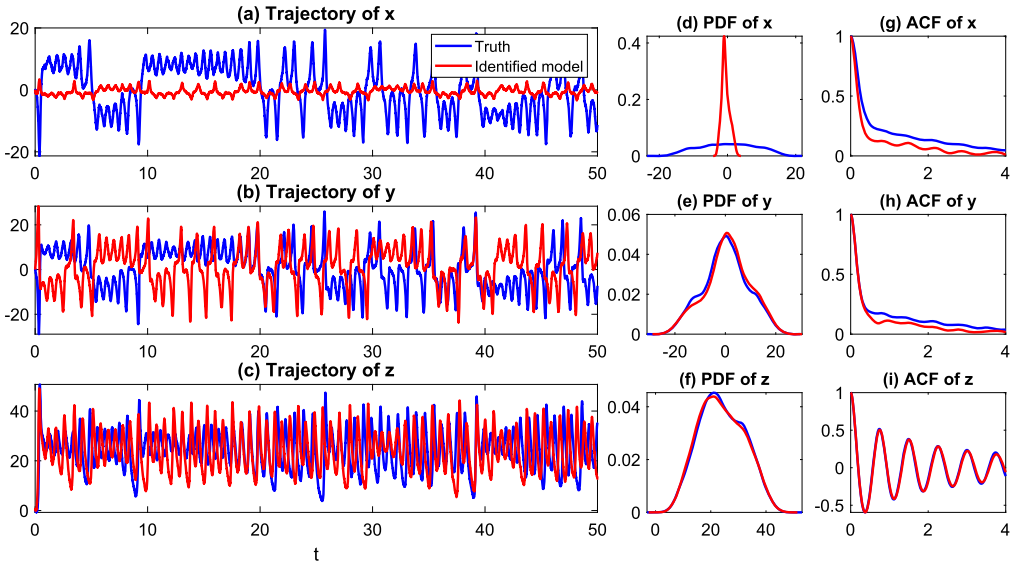

Fig. 10. Comparison of the trajectories (Panels (a)–(c)), the PDFs (Panels (d)–(f)) and the ACFs (Panels (g)–(i)) of the true (blue) and identified model (red), where the true model is the L63 model. The trajectories of both the true and the identified models are simply from one random realization and they are not expected to have one-to-one point-wise correspondence. The true trajectories in Panels (b)–(c) are used as the training data. The ACFs and the PDFs are computed based on long trajectories with 1000 units.

The three parameters σ, ρ and β are the classical choices that result in a butterfly profile of the L63 model. The three noise coefficients provide small noise such that the dynamics has some small scale features with nonlinearity and multiplicative noise while retaining the rough butterfly profile.

Assume the signals of y and z are observed, the model identification based on the general model (45) is shown in Table 2, where the initial values of the parameters are set to be the same as those adopted in Section 7.1. As the test case of the L84 model, the structure of the identified model here is also significantly different from the perfect L63 one. Fig. 10 compares the model trajectories, the PDFs and the ACFs of the perfect L63 model and the identified model. Again, both the dynamical and statistical features associated with the observed variables y and z in the identified model almost perfectly match those in the L63 model. On the other hand, the amplitude of the hidden variable x in the identified model is significantly smaller than that of the L63 model. Thus, the identified model is completely different from the true L63 model but the former is able to generate nearly the perfect dynamical and statistical features as the latter associated with the two observed variables.

7.3. Discussions

The results based on the simple tests in this section convey the following message. In the presence of small-scale noise and partial observations, the model is not guaranteed to be uniquely identified. Different models are able to reproduce all the key features appearing in the observations and provide nearly the same ensemble forecast skill for the observed variables. Yet, the dynamics of the unknown latent or unobserved variables can be quite different. These facts imply that attempting to learn the exact underlying physics of complex turbulent/chaotic systems from purely data-driven methods in the situation with partial observations and (even small) noise is an extremely tough test. Without incorporating suitable physics conditions or constraints into the model development, the model response to external perturbations even for the observed variables can be completely different from the truth despite a high skill of the identified model in capturing the

unperturbed dynamics and statistics. Note that, despite being rare in practice, if certain known perturbations and the corresponding observed model response can be used for model calibration, then the resulting model can be greatly improved. Such a setup with a time-dependent external input also fits into the learning framework developed here.

8. Conclusion

In this article, an efficient learning algorithm based on the EM approach is developed for learning the model and parameters of the conditional Gaussian nonlinear systems with partial observations. The conditional Gaussian nonlinear models are able to capture many important nonlinear and non-Gaussian features as observed in nature. Such a nonlinear modeling framework includes a large class the physics-constrained nonlinear stochastic models, many stochastically coupled reaction-diffusion models in neuroscience and ecology, and quite a few important large-scale dynamical models in turbulence, fluids and geophysical flows. The closed analytic formulae of the nonlinear optimal smoother of the unobserved variables greatly facilitate the EM algorithm. The algorithm only requires a short training data in learning the significant nonlinear and non-Gaussian features. Physics constraint and sparse identification are both incorporated into the learning process while retaining the efficiency of the algorithm. In addition, a judicious block decomposition of both the conditional covariance matrix associated with the nonlinear smoother and the likelihood function facilitates the algorithm to learn the model parameters in many high-dimensional systems with multiscale or multilevel structures. Numerical tests show the skill of the learning algorithm in both perfect model and model error scenarios. The importance of including the physics constraint to prevent the finite-time blowup and the pathological behavior of the model in learning the dynamics is illustrated in a simple example as well. It is also shown that in the presence of noise and partial observations, the model is not guaranteed to be uniquely identified. This indicates that without incorporating suitable physics in the learning process, the identified model may have a different model response to external perturbations, although it perfectly matches the key dynamical and statistical features and obtains nearly the same ensemble forecast skill for the observed variables as the perfect model.

Compared with other learning algorithms, such as the MCMC, the method developed here is much more efficient. In fact, the convergence of the algorithm occurs within only a few iterations and all the manipulations in the algorithm are analytically tractable. The algorithm is also deterministic and it is thus amenable for analysis. The downside of the algorithm is that it may converge to a local optimal solution when the complexity of the underlying system increases. In addition, the current version of the algorithm is an offline method due to the use of the nonlinear smoother technique in recovering the statistics of the hidden variables. Generalizing the algorithm to a real-time fashion can be achieved by combining the basic learning algorithm with a nonlinear filter (Theorem 2.1) or an online version of the nonlinear smoother. A natural issue to explore is how the extreme events in observations advance the online learning process. This remains as a future work.

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

The research of N.C. is supported by the Office of Vice Chancellor for Research and Graduate Education (VCRGE) at University of Wisconsin-Madison and the Office of Naval Research (ONR) MURI N00014-19-1-2421.

Appendix A

A.1. Details of the deriving the solution of the covariance matrix $\mathbf{R}$ *in the M-Step for conditional Gaussian systems*

Recall (16),

$$\mathbf{R} = \frac{1}{J}\left\langle (\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)(\mathbf{Z}^{j+1} - \mathbf{M}^j\boldsymbol{\theta} - \mathbf{C}^j)^*\right\rangle. \tag{48}$$

By defining

$$\widetilde{\mathbf{Z}}^{j+1} = \begin{pmatrix} \mathbf{X}^{j+1} - \mathbf{X}^j \\ \mathbf{Y}^{j+1} - \mathbf{Y}^j \end{pmatrix}, \qquad \text{and} \qquad \widetilde{\mathbf{M}}^j = \begin{pmatrix} \mathbf{A}_\mathbf{0}^j + \mathbf{A}_\mathbf{1}^j\mathbf{Y}^j \\ \mathbf{a}_\mathbf{0}^j + \mathbf{a}_\mathbf{1}^j\mathbf{Y}^j \end{pmatrix}, \tag{49}$$

where $\widetilde{\mathbf{M}}^j$ contains the parameter $\boldsymbol{\theta}$, the equation in (48) becomes

$$\mathbf{R} = \frac{1}{J}\left\langle \widetilde{\mathbf{Z}}^{j+1}(\widetilde{\mathbf{Z}}^{j+1})^* - \widetilde{\mathbf{Z}}^{j+1}(\widetilde{\mathbf{M}}^j)^* - \widetilde{\mathbf{M}}^j(\widetilde{\mathbf{Z}}^{j+1})^* + \widetilde{\mathbf{M}}^j(\widetilde{\mathbf{M}}^j)^*\right\rangle. \tag{50}$$

In order to solve $\mathbf{R}$, the four components on the right hand side of (50) need to be calculated respectively. In light of the definition in (49) and the conditional Gaussian model structure (1), $\langle\widetilde{\mathbf{Z}}^{j+1}(\widetilde{\mathbf{Z}}^{j+1})^*\rangle$ is given by

$$\begin{aligned}\left\langle \widetilde{\mathbf{Z}}^{j+1}(\widetilde{\mathbf{Z}}^{j+1})^*\right\rangle &= \begin{pmatrix} (\mathbf{X}^{j+1}-\mathbf{X}^j)(\mathbf{X}^{j+1}-\mathbf{X}^j)^* & (\mathbf{X}^{j+1}-\mathbf{X}^j)(\mathbf{Y}^{j+1}-\mathbf{Y}^j)^* \\ (\mathbf{Y}^{j+1}-\mathbf{Y}^j)(\mathbf{X}^{j+1}-\mathbf{X}^j)^* & (\mathbf{Y}^{j+1}-\mathbf{Y}^j)(\mathbf{Y}^{j+1}-\mathbf{Y}^j)^* \end{pmatrix} \\ &= \begin{pmatrix} (\mathbf{X}^{j+1}-\mathbf{X}^j)(\mathbf{X}^{j+1}-\mathbf{X}^j)^* & (\mathbf{X}^{j+1}-\mathbf{X}^j)(\boldsymbol{\mu}^{j+1}-\boldsymbol{\mu}^j)^* \\ (\boldsymbol{\mu}^{j+1}-\boldsymbol{\mu}^j)(\mathbf{X}^{j+1}-\mathbf{X}^j)^* & \mathbf{T}_{\mathbf{22}}^{\mathbf{0}} \end{pmatrix},\end{aligned} \tag{51}$$

where

$$\mathbf{T}_{\mathbf{22}}^{\mathbf{0}} = \langle \mathbf{Y}^{j+1}(\mathbf{Y}^{j+1})^*\rangle + \langle \mathbf{Y}^j(\mathbf{Y}^j)^*\rangle - \langle \mathbf{Y}^{j+1}(\mathbf{Y}^j)^*\rangle - \langle \mathbf{Y}^j(\mathbf{Y}^{j+1})^*\rangle. \tag{52}$$

Similarly, an explicit form of $\left\langle \widetilde{\mathbf{Z}}^{j+1}(\widetilde{\mathbf{M}}^j)^*\right\rangle$ satisfies

$$\begin{aligned}\left\langle \widetilde{\mathbf{Z}}^{j+1}(\widetilde{\mathbf{M}}^j)^*\right\rangle &= \begin{pmatrix} \langle(\mathbf{X}^{j+1}-\mathbf{X}^j)(\mathbf{A}_{\mathbf{0}}^j+\mathbf{A}_{\mathbf{1}}^j\mathbf{Y}^j)^*\rangle & \langle(\mathbf{X}^{j+1}-\mathbf{X}^j)(\mathbf{a}_{\mathbf{0}}^j+\mathbf{a}_{\mathbf{1}}^j\mathbf{Y}^j)^*\rangle \\ \langle(\mathbf{Y}^{j+1}-\mathbf{Y}^j)(\mathbf{A}_{\mathbf{0}}^j+\mathbf{A}_{\mathbf{1}}^j\mathbf{Y}^j)^*\rangle & \langle(\mathbf{Y}^{j+1}-\mathbf{Y}^j)(\mathbf{a}_{\mathbf{0}}^j+\mathbf{a}_{\mathbf{1}}^j\mathbf{Y}^j)^*\rangle \end{pmatrix}\Delta t \\ &= \begin{pmatrix} (\mathbf{X}^{j+1}-\mathbf{X}^j)(\mathbf{A}_{\mathbf{0}}^j+\mathbf{A}_{\mathbf{1}}^j\boldsymbol{\mu}^j)^* & (\mathbf{X}^{j+1}-\mathbf{X}^j)(\mathbf{a}_{\mathbf{0}}^j+\mathbf{a}_{\mathbf{1}}^j\boldsymbol{\mu}^j)^* \\ \mathbf{T}_{\mathbf{21}}^{\mathbf{1}} & \mathbf{T}_{\mathbf{22}}^{\mathbf{1}} \end{pmatrix}\Delta t\end{aligned} \tag{53}$$

where

$$\begin{aligned}\mathbf{T}_{\mathbf{21}}^{\mathbf{1}} &= (\boldsymbol{\mu}^{j+1}-\boldsymbol{\mu}^j)(\mathbf{A}_{\mathbf{0}}^j)^* + \langle \mathbf{Y}^{j+1}(\mathbf{Y}^j)^*\rangle(\mathbf{A}_{\mathbf{1}}^j)^* - \langle \mathbf{Y}^j(\mathbf{Y}^j)^*\rangle(\mathbf{A}_{\mathbf{1}}^j)^* \\ \mathbf{T}_{\mathbf{22}}^{\mathbf{1}} &= (\boldsymbol{\mu}^{j+1}-\boldsymbol{\mu}^j)(\mathbf{a}_{\mathbf{0}}^j)^* + \langle \mathbf{Y}^{j+1}(\mathbf{Y}^j)^*\rangle(\mathbf{a}_{\mathbf{1}}^j)^* - \langle \mathbf{Y}^j(\mathbf{Y}^j)^*\rangle(\mathbf{a}_{\mathbf{1}}^j)^*\end{aligned} \tag{54}$$

Finally, the last term $\left\langle \widetilde{\mathbf{M}}^j(\widetilde{\mathbf{M}}^j)^*\right\rangle$ is given by

$$\begin{aligned}\left\langle \widetilde{\mathbf{M}}^j(\widetilde{\mathbf{M}}^j)^*\right\rangle &= \begin{pmatrix} \langle(\mathbf{A}_{\mathbf{0}}^j+\mathbf{A}_{\mathbf{1}}^j\mathbf{Y}^j)(\mathbf{A}_{\mathbf{0}}^j+\mathbf{A}_{\mathbf{1}}^j\mathbf{Y}^j)^*\rangle & \langle(\mathbf{A}_{\mathbf{0}}^j+\mathbf{A}_{\mathbf{1}}^j\mathbf{Y}^j)(\mathbf{a}_{\mathbf{0}}^j+\mathbf{a}_{\mathbf{1}}^j\mathbf{Y}^j)^*\rangle \\ \langle(\mathbf{a}_{\mathbf{0}}^j+\mathbf{a}_{\mathbf{1}}^j\mathbf{Y}^j)(\mathbf{A}_{\mathbf{0}}^j+\mathbf{A}_{\mathbf{1}}^j\mathbf{Y}^j)^*\rangle & \langle(\mathbf{a}_{\mathbf{0}}^j+\mathbf{a}_{\mathbf{1}}^j\mathbf{Y}^j)(\mathbf{a}_{\mathbf{0}}^j+\mathbf{a}_{\mathbf{1}}^j\mathbf{Y}^j)^*\rangle \end{pmatrix}(\Delta t)^2 \\ &= \begin{pmatrix} \mathbf{T}_{\mathbf{11}}^{\mathbf{2}} & \mathbf{T}_{\mathbf{12}}^{\mathbf{2}} \\ \mathbf{T}_{\mathbf{21}}^{\mathbf{2}} & \mathbf{T}_{\mathbf{22}}^{\mathbf{2}} \end{pmatrix}(\Delta t)^2\end{aligned} \tag{55}$$

where

$$\begin{aligned}\mathbf{T}_{\mathbf{11}}^{\mathbf{2}} &= \mathbf{A}_{\mathbf{0}}^j(\mathbf{A}_{\mathbf{0}}^j)^* + \mathbf{A}_{\mathbf{0}}^j\langle(\mathbf{Y}^j)^*\rangle(\mathbf{A}_{\mathbf{1}}^j)^* + \mathbf{A}_{\mathbf{1}}^j\langle\mathbf{Y}^j\rangle(\mathbf{A}_{\mathbf{0}}^j)^* + \mathbf{A}_{\mathbf{1}}^j\langle\mathbf{Y}^j(\mathbf{Y}^j)^*\rangle(\mathbf{A}_{\mathbf{1}}^j)^* \\ \mathbf{T}_{\mathbf{12}}^{\mathbf{2}} &= \mathbf{A}_{\mathbf{0}}^j(\mathbf{a}_{\mathbf{0}}^j)^* + \mathbf{A}_{\mathbf{0}}^j\langle(\mathbf{Y}^j)^*\rangle(\mathbf{a}_{\mathbf{1}}^j)^* + \mathbf{A}_{\mathbf{1}}^j\langle\mathbf{Y}^j\rangle(\mathbf{a}_{\mathbf{0}}^j)^* + \mathbf{A}_{\mathbf{1}}^j\langle\mathbf{Y}^j(\mathbf{Y}^j)^*\rangle(\mathbf{a}_{\mathbf{1}}^j)^* \\ \mathbf{T}_{\mathbf{21}}^{\mathbf{2}} &= \mathbf{a}_{\mathbf{0}}^j(\mathbf{A}_{\mathbf{0}}^j)^* + \mathbf{a}_{\mathbf{0}}^j\langle(\mathbf{Y}^j)^*\rangle(\mathbf{A}_{\mathbf{1}}^j)^* + \mathbf{a}_{\mathbf{1}}^j\langle\mathbf{Y}^j\rangle(\mathbf{A}_{\mathbf{0}}^j)^* + \mathbf{a}_{\mathbf{1}}^j\langle\mathbf{Y}^j(\mathbf{Y}^j)^*\rangle(\mathbf{A}_{\mathbf{1}}^j)^* \\ \mathbf{T}_{\mathbf{22}}^{\mathbf{2}} &= \mathbf{a}_{\mathbf{0}}^j(\mathbf{a}_{\mathbf{0}}^j)^* + \mathbf{a}_{\mathbf{0}}^j\langle(\mathbf{Y}^j)^*\rangle(\mathbf{a}_{\mathbf{1}}^j)^* + \mathbf{a}_{\mathbf{1}}^j\langle\mathbf{Y}^j\rangle(\mathbf{a}_{\mathbf{0}}^j)^* + \mathbf{a}_{\mathbf{1}}^j\langle\mathbf{Y}^j(\mathbf{Y}^j)^*\rangle(\mathbf{a}_{\mathbf{1}}^j)^*\end{aligned} \tag{56}$$

To compute (54) and (56), the quadratic terms of $\mathbf{Y}$ need to be solved,

$$\begin{aligned}\langle \mathbf{Y}^{j+1}(\mathbf{Y}^{j+1})^*\rangle &= \left\langle(\mathbf{Y}^{j+1}-\langle\mathbf{Y}^{j+1}\rangle+\langle\mathbf{Y}^{j+1}\rangle)(\mathbf{Y}^{j+1}-\langle\mathbf{Y}^{j+1}\rangle+\langle\mathbf{Y}^{j+1}\rangle)\right\rangle \\ &= \mathbf{R}_{\mathbf{s}}^{j+1} + \boldsymbol{\mu}_{\mathbf{s}}^{j+1}(\boldsymbol{\mu}_{\mathbf{s}}^{j+1})^* \\ \langle \mathbf{Y}^j(\mathbf{Y}^j)^*\rangle &= \left\langle(\mathbf{Y}^j-\langle\mathbf{Y}^j\rangle+\langle\mathbf{Y}^j\rangle)(\mathbf{Y}^j-\langle\mathbf{Y}^j\rangle+\langle\mathbf{Y}^j\rangle)\right\rangle \\ &= \mathbf{R}_{\mathbf{s}}^j + \boldsymbol{\mu}_{\mathbf{s}}^j(\boldsymbol{\mu}_{\mathbf{s}}^j)^* \\ \langle \mathbf{Y}^{j+1}(\mathbf{Y}^j)^*\rangle &= \left\langle(\mathbf{Y}^{j+1}-\langle\mathbf{Y}^{j+1}\rangle+\langle\mathbf{Y}^{j+1}\rangle)(\mathbf{Y}^j-\langle\mathbf{Y}^j\rangle+\langle\mathbf{Y}^j\rangle)\right\rangle \\ &= \mathbf{R}_{\mathbf{s}}^{j+1}(\mathbf{C}^j)^* + \boldsymbol{\mu}_{\mathbf{s}}^{j+1}(\boldsymbol{\mu}_{\mathbf{s}}^j)^* \\ \mathbf{C}^j &= \mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_{\mathbf{1}}^j\Delta t)^*(\mathbf{b}_{\mathbf{Y}}^j(\mathbf{b}_{\mathbf{Y}}^j)^*\Delta t + (\mathbf{I}+\mathbf{a}_{\mathbf{1}}^j\Delta t)\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_{\mathbf{1}}^j\Delta t)^*)^{-1},\end{aligned} \tag{57}$$

where the details of computing the temporal cross-covariance term $\langle \mathbf{Y}^{j+1}(\mathbf{Y}^j)^*\rangle$ is shown in Theorem A.2 in Section A.2.

A.2. Temporal cross-covariance in the smoother estimate

This subsection aims at finding an explicit formula of computing the temporal cross-covariance term $\langle \mathbf{Y}^{j+1}(\mathbf{Y}^j)^*\rangle = \mathbf{R}_{\mathbf{s}}^{j+1,j} + \boldsymbol{\mu}_{\mathbf{s}}^{j+1}(\boldsymbol{\mu}_{\mathbf{s}}^j)^*$ or equivalently $\mathbf{R}_{\mathbf{s}}^{j+1,j}$.

Lemma A.1 *(Matrix inversion formulae).*

$$\begin{aligned}(\mathbf{A}+\mathbf{BCD})^{-1} &= \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{C}^{-1}+\mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1} \\ \left(\mathbf{I}-(\mathbf{A}+\mathbf{B})^{-1}\mathbf{A}\right)\mathbf{B}^{-1} &= (\mathbf{A}+\mathbf{B})^{-1}\end{aligned} \tag{58}$$

Theorem A.2. *The temporal cross-covariance is given by*

$$\begin{aligned}\langle \mathbf{Y}^{j+1}(\mathbf{Y}^{j})^{*}\rangle &= \left\langle (\mathbf{Y}^{j+1}-\langle \mathbf{Y}^{j+1}\rangle + \langle \mathbf{Y}^{j+1}\rangle)(\mathbf{Y}^{j}-\langle \mathbf{Y}^{j}\rangle + \langle \mathbf{Y}^{j}\rangle)\right\rangle \\ &= \mathbf{R}_{\mathbf{s}}^{j+1}(\mathbf{C}^{j})^{*} + \boldsymbol{\mu}_{\mathbf{s}}^{j+1}(\boldsymbol{\mu}_{\mathbf{s}}^{j})^{*},\end{aligned} \tag{59}$$

where

$$\mathbf{C}^{j} = \mathbf{R}_{\mathbf{f}}^{j}(\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)^{*}(\mathbf{b}_{\mathbf{Y}}^{j}(\mathbf{b}_{\mathbf{Y}}^{j})^{*}\Delta t + (\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)\mathbf{R}_{\mathbf{f}}^{j}(\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)^{*})^{-1}.$$

Proof. Consider the logarithm of the conditional joint distribution,

$$\log p(\mathbf{Y}^{j+1},\mathbf{Y}^{j}|\mathbf{X}^{s}, s\le J) = \log p(\mathbf{Y}^{j}|\mathbf{Y}^{j+1},\mathbf{X}^{s}, s\le j) + \log p(\mathbf{Y}^{j+1}|\mathbf{X}^{s}, s\le J), \tag{60}$$

where the Markov property has been applied such that J is replaced by j in the first term on the right hand side. Writing into a more detailed form, the above equation leads to

$$\begin{aligned}&\log p(\mathbf{Y}^{j+1},\mathbf{Y}^{j}|\mathbf{X}^{s}, s\le J) \\ &\quad= \log p(\mathbf{Y}^{j}|\mathbf{Y}^{j+1},\mathbf{X}^{s}, s\le j) + \log p(\mathbf{Y}^{j+1}|\mathbf{X}^{s}, s\le J) \\ &\quad= \log p(\mathbf{Y}^{j+1}|\mathbf{Y}^{j}) + \log p(\mathbf{Y}^{j}|\mathbf{X}^{s}, s\le j) - \log p(\mathbf{Y}^{j+1}|\mathbf{X}^{s}, s\le j) + \log p(\mathbf{Y}^{j+1}|\mathbf{X}^{s}, s\le J) \\ &\quad= -\frac{1}{2}\left[\mathbf{Y}^{j+1} - \left(\mathbf{a}_{\mathbf{0}}^{j}\Delta t + (\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)\mathbf{Y}^{j}\right)\right]^{*}(\mathbf{b}_{\mathbf{Y}}\mathbf{b}_{\mathbf{Y}}^{*}\Delta t)^{-1}\left[\mathbf{Y}^{j+1} - \left(\mathbf{a}_{\mathbf{0}}^{j}\Delta t + (\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)\mathbf{Y}^{j}\right)\right] \\ &\qquad+ \frac{1}{2}\left[\mathbf{Y}^{j+1} - \left(\mathbf{a}_{\mathbf{0}}^{j}\Delta t + (\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)\boldsymbol{\mu}^{j}\right)\right]^{*}(\mathbf{b}_{\mathbf{Y}}\mathbf{b}_{\mathbf{Y}}^{*}\Delta t)^{-1}\left[\mathbf{Y}^{j+1} - \left(\mathbf{a}_{\mathbf{0}}^{j}\Delta t + (\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)\boldsymbol{\mu}^{j}\right)\right] \\ &\qquad- \frac{1}{2}(\mathbf{Y}^{j}-\boldsymbol{\mu}^{j})^{*}(\mathbf{R}_{\mathbf{f}}^{j})^{-1}(\mathbf{Y}^{j}-\boldsymbol{\mu}^{j}) - \frac{1}{2}(\mathbf{Y}^{j+1}-\boldsymbol{\mu}_{\mathbf{s}}^{j+1})^{*}(\mathbf{R}_{\mathbf{s}}^{j+1})^{-1}(\mathbf{Y}^{j+1}-\boldsymbol{\mu}_{\mathbf{s}}^{j+1}) \\ &\quad= -\frac{1}{2}(\mathbf{Y}^{j+1})^{*}\left[(\mathbf{b}_{\mathbf{Y}}\mathbf{b}_{\mathbf{Y}}^{*}\Delta t)^{-1} - (\mathbf{b}_{\mathbf{Y}}\mathbf{b}_{\mathbf{Y}}^{*}\Delta t)^{-1} + (\mathbf{R}_{\mathbf{s}}^{j+1})^{-1}\right]\mathbf{Y}^{j+1} \\ &\qquad- \frac{1}{2}(\mathbf{Y}^{j+1})^{*}\left[(-\mathbf{b}_{\mathbf{Y}}\mathbf{b}_{\mathbf{Y}}^{*}\Delta t)^{-1}(\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)\right]\mathbf{Y}^{j} - \frac{1}{2}(\mathbf{Y}^{j})^{*}\left[(\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)^{*}(\mathbf{b}_{\mathbf{Y}}\mathbf{b}_{\mathbf{Y}}^{*}\Delta t)^{-1}\right](\mathbf{Y}^{j+1}) \\ &\qquad- \frac{1}{2}(\mathbf{Y}^{j})^{*}\left[(\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)^{*}(\mathbf{b}_{\mathbf{Y}}\mathbf{b}_{\mathbf{Y}}^{*}\Delta t)^{-1}(\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t) + (\mathbf{R}_{\mathbf{f}}^{j})^{-1}\right]\mathbf{Y}^{j} + (\mathbf{Y}^{j})^{*}(\mathbf{R}_{\mathbf{f}}^{j})^{-1}\boldsymbol{\mu}^{j} + \ldots\end{aligned} \tag{61}$$

On the other hand, for a joint Gaussian distribution, the following equality is valid,

$$\begin{aligned}\log p(\mathbf{z}_{\mathbf{1}},\mathbf{z}_{\mathbf{2}}) &= -\frac{1}{2}\begin{pmatrix}\mathbf{z}_{\mathbf{1}}-\boldsymbol{\mu}_{\mathbf{1}}\\ \mathbf{z}_{\mathbf{2}}-\boldsymbol{\mu}_{\mathbf{2}}\end{pmatrix}^{*}\begin{pmatrix}\mathbf{S}_{\mathbf{11}} & \mathbf{S}_{\mathbf{12}}\\ \mathbf{S}_{\mathbf{21}} & \mathbf{S}_{\mathbf{22}}\end{pmatrix}\begin{pmatrix}\mathbf{z}_{\mathbf{1}}-\boldsymbol{\mu}_{\mathbf{1}}\\ \mathbf{z}_{\mathbf{2}}-\boldsymbol{\mu}_{\mathbf{2}}\end{pmatrix} + \ldots \\ &= -\frac{1}{2}\mathbf{z}_{\mathbf{1}}^{*}\mathbf{S}_{\mathbf{11}}\mathbf{z}_{\mathbf{1}} - \frac{1}{2}\mathbf{z}_{\mathbf{1}}^{*}\mathbf{S}_{\mathbf{12}}\mathbf{z}_{\mathbf{2}} - \frac{1}{2}\mathbf{z}_{\mathbf{2}}^{*}\mathbf{S}_{\mathbf{21}}\mathbf{z}_{\mathbf{1}} - \frac{1}{2}\mathbf{z}_{\mathbf{2}}^{*}\mathbf{S}_{\mathbf{22}}\mathbf{z}_{\mathbf{2}} + \mathbf{z}_{\mathbf{2}}^{*}(\mathbf{S}_{\mathbf{21}}\boldsymbol{\mu}_{\mathbf{1}} + \mathbf{S}_{\mathbf{22}}\boldsymbol{\mu}_{\mathbf{2}}) + \ldots\end{aligned} \tag{62}$$

The covariance matrix is

$$\begin{aligned}\begin{pmatrix}\boldsymbol{\Sigma}_{\mathbf{11}} & \boldsymbol{\Sigma}_{\mathbf{12}}\\ \boldsymbol{\Sigma}_{\mathbf{21}} & \boldsymbol{\Sigma}_{\mathbf{22}}\end{pmatrix} &= \begin{pmatrix}\mathbf{S}_{\mathbf{11}} & \mathbf{S}_{\mathbf{12}}\\ \mathbf{S}_{\mathbf{21}} & \mathbf{S}_{\mathbf{22}}\end{pmatrix}^{-1} = \begin{pmatrix}\mathbf{F}_{\mathbf{11}}^{-1} & -\mathbf{F}_{\mathbf{11}}^{-1}\mathbf{S}_{\mathbf{12}}\mathbf{S}_{\mathbf{22}}^{-1}\\ -\mathbf{S}_{\mathbf{22}}^{-1}\mathbf{S}_{\mathbf{21}}\mathbf{F}_{\mathbf{11}}^{-1} & \mathbf{F}_{\mathbf{22}}^{-1}\end{pmatrix} \\ &= \begin{pmatrix}\mathbf{S}_{\mathbf{11}}^{-1}+\mathbf{S}_{\mathbf{11}}^{-1}\mathbf{S}_{\mathbf{12}}\mathbf{F}_{\mathbf{22}}^{-1}\mathbf{S}_{\mathbf{21}}\mathbf{S}_{\mathbf{11}}^{-1} & -\mathbf{F}_{\mathbf{11}}^{-1}\mathbf{S}_{\mathbf{12}}\mathbf{S}_{\mathbf{22}}^{-1}\\ -\mathbf{S}_{\mathbf{22}}^{-1}\mathbf{S}_{\mathbf{21}}\mathbf{F}_{\mathbf{11}}^{-1} & \mathbf{S}_{\mathbf{22}}^{-1}+\mathbf{S}_{\mathbf{22}}^{-1}\mathbf{S}_{\mathbf{21}}\mathbf{F}_{\mathbf{11}}^{-1}\mathbf{S}_{\mathbf{12}}\mathbf{S}_{\mathbf{22}}^{-1}\end{pmatrix},\end{aligned} \tag{63}$$

where

$$\mathbf{F}_{\mathbf{11}} = \mathbf{S}_{\mathbf{11}} - \mathbf{S}_{\mathbf{12}}\mathbf{S}_{\mathbf{22}}^{-1}\mathbf{S}_{\mathbf{21}} \qquad \text{and} \qquad \mathbf{F}_{\mathbf{22}} = \mathbf{S}_{\mathbf{22}} - \mathbf{S}_{\mathbf{21}}\mathbf{S}_{\mathbf{11}}^{-1}\mathbf{S}_{\mathbf{21}}. \tag{64}$$

Regarding $\mathbf{z}_{\mathbf{1}} = \mathbf{Y}^{j+1}$ and $\mathbf{z}_{\mathbf{2}} = \mathbf{Y}^{j}$, making a link between (61) and (62) leads to

$$\begin{aligned}&\begin{pmatrix}\mathbf{R}_{\mathbf{s}}^{j+1} & \mathbf{R}_{\mathbf{s}}^{j+1,j}\\ \mathbf{R}_{\mathbf{s}}^{j,j+1} & \mathbf{R}_{\mathbf{s}}^{j}\end{pmatrix} = \begin{pmatrix}\mathbf{S}_{\mathbf{11}} & \mathbf{S}_{\mathbf{12}}\\ \mathbf{S}_{\mathbf{21}} & \mathbf{S}_{\mathbf{22}}\end{pmatrix}^{-1} \\ &= \begin{pmatrix}(\mathbf{R}_{\mathbf{s}}^{j+1})^{-1} & (-\mathbf{b}_{\mathbf{Y}}\mathbf{b}_{\mathbf{Y}}^{*}\Delta t)^{-1}(\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)\\ -(\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)^{*}(\mathbf{b}_{\mathbf{Y}}\mathbf{b}_{\mathbf{Y}}^{*}\Delta t)^{-1} & (\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)^{*}(\mathbf{b}_{\mathbf{Y}}\mathbf{b}_{\mathbf{Y}}^{*}\Delta t)^{-1}(\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t) + (\mathbf{R}_{\mathbf{f}}^{j})^{-1}\end{pmatrix}^{-1}\end{aligned} \tag{65}$$

From (65), it is clear that $\mathbf{S}_{22}$ yields,

$$\mathbf{S}_{22} = (\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t)^{*}(\mathbf{b}_{\mathbf{Y}}^{j}(\mathbf{b}_{\mathbf{Y}}^{j})^{*}\Delta t)^{-1}(\mathbf{I}+\mathbf{a}_{\mathbf{1}}^{j}\Delta t) + (\mathbf{R}_{\mathbf{f}}^{j})^{-1}. \tag{66}$$

In light of Lemma A.1, (66) leads to

$$\begin{aligned}\mathbf{S}_{22}^{-1} &= \left[(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*(\mathbf{b}_{\mathbf{Y}}^j(\mathbf{b}_{\mathbf{Y}}^j)^*\Delta t)^{-1}(\mathbf{I}+\mathbf{a}_1^j\Delta t)+(\mathbf{R}_{\mathbf{f}}^j)^{-1}\right]^{-1}\\ &= \mathbf{R}_{\mathbf{f}}^j-\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*(\mathbf{b}_{\mathbf{Y}}^j(\mathbf{b}_{\mathbf{Y}}^j)^*\Delta t+(\mathbf{I}+\mathbf{a}_1^j\Delta t)\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*)^{-1}(\mathbf{I}+\mathbf{a}_1^j\Delta t)\mathbf{R}_{\mathbf{f}}^j\\ &= \mathbf{R}_{\mathbf{f}}^j-\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*(\mathbf{P}^{j+1})^{-1}(\mathbf{I}+\mathbf{a}_1^j\Delta t)\mathbf{R}_{\mathbf{f}}^j\\ &:= \mathbf{R}_{\mathbf{f}}^j-\mathbf{C}^j\mathbf{P}^{j+1}(\mathbf{C}^j)^*,\end{aligned} \tag{67}$$

where

$$\begin{aligned}\mathbf{P}^{j+1} &= \mathbf{b}_{\mathbf{Y}}^j(\mathbf{b}_{\mathbf{Y}}^j)^*\Delta t+(\mathbf{I}+\mathbf{a}_1^j\Delta t)\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*,\\ \mathbf{C}^j &= \mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*(\mathbf{P}^{j+1})^{-1}.\end{aligned}$$

Next, in light of (65) and (67), we have

$$\begin{aligned}\mathbf{S}_{22}^{-1}\mathbf{S}_{21} &= -\left[\mathbf{R}_{\mathbf{f}}^j-\mathbf{C}^j\mathbf{P}^{j+1}(\mathbf{C}^j)^*\right](\mathbf{I}+\mathbf{a}_1^j\Delta t)^*(\mathbf{b}_{\mathbf{Y}}^j(\mathbf{b}_{\mathbf{Y}}^j)^*\Delta t)^{-1}\\ &= -\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*\left[\mathbf{I}-(\mathbf{I}+\mathbf{a}_1^j\Delta t)^{-*}(\mathbf{R}_{\mathbf{f}}^j)^{-1}\mathbf{C}^j\mathbf{P}^{j+1}(\mathbf{C}^j)^*(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*\right](\mathbf{b}_{\mathbf{Y}}^j(\mathbf{b}_{\mathbf{Y}}^j)^*\Delta t)^{-1}\\ &= -\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*\Big(\mathbf{I}-(\mathbf{I}+\mathbf{a}_1^j\Delta t)^{-*}(\mathbf{R}_{\mathbf{f}}^j)^{-1}\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*(\mathbf{P}^{j+1})^{-1}\mathbf{P}^{j+1}\times\\ &\qquad(\mathbf{P}^{j+1})^*(\mathbf{I}+\mathbf{a}_1^j\Delta t)\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)\Big)(\mathbf{b}_{\mathbf{Y}}^j(\mathbf{b}_{\mathbf{Y}}^j)^*\Delta t)^{-1}\\ &= -\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*\Big(\mathbf{I}-(\mathbf{P}^{j+1})^{-*}(\mathbf{I}+\mathbf{a}_1^j\Delta t)\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*\Big)(\mathbf{b}_{\mathbf{Y}}^j(\mathbf{b}_{\mathbf{Y}}^j)^*\Delta t)^{-1}\\ &= -\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*\Big[\mathbf{I}-\Big((\mathbf{I}+\mathbf{a}_1^j\Delta t)\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*+\mathbf{b}_{\mathbf{Y}}^j(\mathbf{b}_{\mathbf{Y}}^j)^*\Delta t\Big)\times\\ &\qquad(\mathbf{I}+\mathbf{a}_1^j\Delta t)\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*\Big](\mathbf{b}_{\mathbf{Y}}^j(\mathbf{b}_{\mathbf{Y}}^j)^*\Delta t)^{-1}\\ &= -\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*\Big((\mathbf{I}+\mathbf{a}_1^j\Delta t)\mathbf{R}_{\mathbf{f}}^j(\mathbf{I}+\mathbf{a}_1^j\Delta t)^*+\mathbf{b}_{\mathbf{Y}}^j(\mathbf{b}_{\mathbf{Y}}^j)^*\Delta t\Big)^{-1}\\ &= -\mathbf{C}^j,\end{aligned} \tag{68}$$

where (58) in Lemma A.1 has been applied.

$$\begin{aligned}\mathbf{R}_{\mathbf{s}}^j &= \mathbf{S}_{22}^{-1}+\mathbf{S}_{22}^{-1}\mathbf{S}_{21}\mathbf{F}_{11}^{-1}\mathbf{S}_{12}\mathbf{S}_{22}^{-1}\\ &= \left(\mathbf{R}_{\mathbf{f}}^j-\mathbf{C}^j\mathbf{P}^{j+1}(\mathbf{C}^j)^*\right)+\mathbf{C}^j\mathbf{R}_{\mathbf{s}}^{j+1}(\mathbf{C}^j)^*\\ &= \mathbf{R}_{\mathbf{f}}^j+\mathbf{C}^j(\mathbf{R}_{\mathbf{s}}^{j+1}-\mathbf{P}^{j+1})(\mathbf{C}^j)^*,\end{aligned} \tag{69}$$

where we have used the fact that $\mathbf{F}_{\mathbf{11}}^{-1}=\mathbf{R}_{\mathbf{s}}^{j+1}$. Finally we have

$$\mathbf{R}_{\mathbf{s}}^{j+1,j} = -\mathbf{F}_{\mathbf{11}}^{-1}\mathbf{S}_{\mathbf{12}}\mathbf{S}_{\mathbf{22}}^{-1} = \mathbf{R}_{\mathbf{s}}^{j+1}(\mathbf{C}^j)^*. \tag{70}$$

This finishes the proof. □

A.3. A numerical trick of accelerating the learning process

In some situations, the learning process may have a slow convergence. The following numerical trick can accelerate the convergence of the algorithm. Denote $\boldsymbol{\theta}^{(k-1)}$ and $\boldsymbol{\theta}^{(k)}$ the learned parameters (or part of the learned parameters) in the previous $(k-1)$ and current (k) iterations. An acceleration of the parameter value at the current step can be achieved by

$$\boldsymbol{\theta}_{new}^{(k)} = \boldsymbol{\theta}^{(k-1)}+\alpha(\boldsymbol{\theta}^{(k)}-\boldsymbol{\theta}^{(k-1)}), \tag{71}$$

where α is a hyper parameter. If $\alpha=1$, then there is no acceleration. The acceleration rate depends on the amplitude of α. Such a trick can be applied in the first a few iterations especially for those in the unobserved processes when the observability of the system is weak.

References

[1] A.J. Majda, Introduction to Turbulent Dynamical Systems in Complex Systems, Springer, 2016.
[2] S.H. Strogatz, Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering, CRC Press, 2018.
[3] D. Baleanu, J.A.T. Machado, A.C. Luo, Fractional Dynamics and Control, Springer Science & Business Media, 2011.
[4] T. Deisboeck, J.Y. Kresh, Complex Systems Science in Biomedicine, Springer Science & Business Media, 2007.
[5] E. Kalnay, Atmospheric Modeling, Data Assimilation and Predictability, Cambridge University Press, 2003.
[6] W. Lahoz, B. Khattatov, R. Ménard, Data assimilation and information, in: Data Assimilation, Springer, 2010, pp. 3–12.
[7] A.J. Majda, J. Harlim, Filtering Complex Turbulent Systems, Cambridge University Press, 2012.
[8] G. Evensen, Data Assimilation: the Ensemble Kalman Filter, Springer Science & Business Media, 2009.
[9] K. Law, A. Stuart, K. Zygalakis, Data Assimilation: a Mathematical Introduction, vol. 62, Springer, 2015.
[10] M. Farazmand, T. Sapsis, Extreme events: mechanisms and prediction, Appl. Mech. Rev. 71 (5) (2019).
[11] M.W. Denny, L.J. Hunt, L.P. Miller, C.D. Harley, On the prediction of extreme ecological events, Ecol. Monogr. 79 (3) (2009) 397–421.
[12] M.A. Mohamad, T.P. Sapsis, Probabilistic description of extreme events in intermittently unstable dynamical systems excited by correlated stochastic processes, SIAM/ASA J. Uncertain. Quantificat. 3 (1) (2015) 709–736.
[13] T.N. Palmer, A nonlinear dynamical perspective on model error: a proposal for non-local stochastic-dynamic parametrization in weather and climate prediction models, Q. J. R. Meteorol. Soc. 127 (572) (2001) 279–304.
[14] A.J. Majda, M. Branicki, Lessons in uncertainty quantification for turbulent dynamical systems, Discrete Contin. Dyn. Syst., Ser. A 32 (9) (2012) 3133–3221.
[15] D. Orrell, L. Smith, J. Barkmeijer, T. Palmer, Model error in weather forecasting, Nonlinear Process. Geophys. 8 (6) (2001) 357–371.
[16] X.-M. Hu, F. Zhang, J.W. Nielsen-Gammon, Ensemble-based simultaneous state and parameter estimation for treatment of mesoscale model error: a real-data study, Geophys. Res. Lett. 37 (8) (2010).
[17] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM Rev. 57 (4) (2015) 483–531.
[18] Z. Ghahramani, G.E. Hinton, Parameter estimation for linear dynamical systems, Technical Report CRG-TR-96-2, University of Toronto, Dept. of Computer Science, 1996.
[19] R.H. Shumway, D.S. Stoffer, An approach to time series smoothing and forecasting using the EM algorithm, J. Time Ser. Anal. 3 (4) (1982) 253–264.
[20] C. Andrieu, N. De Freitas, A. Doucet, M.I. Jordan, An introduction to MCMC for machine learning, Mach. Learn. 50 (1–2) (2003) 5–43.
[21] W.H. Press, S.A. Teukolsky, B.P. Flannery, W.T. Vetterling, Numerical Recipes with Source Code CD-ROM 3rd Edition: The Art of Scientific Computing, Cambridge University Press, 2007.
[22] M. Richey, The evolution of Markov chain Monte Carlo methods, Am. Math. Mon. 117 (5) (2010) 383–413.
[23] H. Haario, M. Laine, A. Mira, E. Saksman, Dram: efficient adaptive MCMC, Stat. Comput. 16 (4) (2006) 339–354.
[24] S. Chib, E. Greenberg, Y. Chen, et al., MCMC methods for fitting and comparing multinomial response models, Economics Working Paper Archive, Econometrics 9802001, 1998.
[25] M.A. Tanner, W.H. Wong, The calculation of posterior distributions by data augmentation, J. Am. Stat. Assoc. 82 (398) (1987) 528–540.
[26] A. Golightly, D.J. Wilkinson, Bayesian inference for nonlinear multivariate diffusion models observed with error, Comput. Stat. Data Anal. 52 (3) (2008) 1674–1693.
[27] G.C. Wei, M.A. Tanner, A Monte Carlo implementation of the em algorithm and the poor man's data augmentation algorithms, J. Am. Stat. Assoc. 85 (411) (1990) 699–704.
[28] C. Andrieu, G.O. Roberts, et al., The pseudo-marginal approach for efficient Monte Carlo computations, Ann. Stat. 37 (2) (2009) 697–725.
[29] O. Stramer, M. Bognar, et al., Bayesian inference for irreducible diffusion processes using the pseudo-marginal approach, Bayesian Anal. 6 (2) (2011) 231–258.
[30] S. Särkkä, Bayesian Filtering and Smoothing, vol. 3, Cambridge University Press, 2013.
[31] O.M. Smedstad, J.J. O'Brien, Variational data assimilation and parameter estimation in an equatorial Pacific Ocean model, Prog. Oceanogr. 26 (2) (1991) 179–241.
[32] D.P. Dee, On-line estimation of error covariance parameters for atmospheric data assimilation, Mon. Weather Rev. 123 (4) (1995) 1128–1145.
[33] M. Rodriguez-Fernandez, P. Mendes, J.R. Banga, A hybrid approach for efficient and robust parameter estimation in biochemical pathways, Biosystems 83 (2–3) (2006) 248–265.
[34] K. Schittkowski, Numerical Data Fitting in Dynamical Systems: a Practical Introduction with Applications and Software, vol. 77, Springer Science & Business Media, 2013.
[35] N. Chen, A. Majda, Conditional Gaussian systems for multiscale nonlinear stochastic systems: prediction, state estimation and uncertainty quantification, Entropy 20 (7) (2018) 509.
[36] N. Chen, A.J. Majda, Filtering nonlinear turbulent dynamical systems through conditional Gaussian statistics, Mon. Weather Rev. 144 (12) (2016) 4885–4917.
[37] A.J. Majda, J. Harlim, Physics constrained nonlinear regression models for time series, Nonlinearity 26 (1) (2012) 201.
[38] J. Harlim, A. Mahdi, A.J. Majda, An ensemble Kalman filter for statistical estimation of physics constrained nonlinear regression models, J. Comput. Phys. 257 (2014) 782–812.
[39] B. Lindner, J. Garcıa-Ojalvo, A. Neiman, L. Schimansky-Geier, Effects of noise in excitable systems, Phys. Rep. 392 (6) (2004) 321–424.
[40] A.B. Medvinsky, S.V. Petrovskii, I.A. Tikhonova, H. Malchow, B.-L. Li, Spatiotemporal complexity of plankton and fish dynamics, SIAM Rev. 44 (3) (2002) 311–370.
[41] R. Salmon, Lectures on Geophysical Fluid Dynamics, Oxford University Press, 1998.
[42] G.K. Vallis, Atmospheric and Oceanic Fluid Dynamics, Cambridge University Press, 2017.
[43] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc., Ser. B, Methodol. 39 (1) (1977) 1–22.
[44] N. Chen, A.J. Majda, Efficient nonlinear optimal smoothing and sampling algorithms for complex turbulent nonlinear dynamical systems with partial observations, J. Comput. Phys. (2020) 109381, https://doi.org/10.1016/j.jcp.2020.109381.
[45] Z. Ghahramani, S.T. Roweis, Learning nonlinear dynamical systems using an em algorithm, in: Advances in Neural Information Processing Systems, 1999, pp. 431–437.
[46] A.J. Majda, Y. Yuan, Fundamental limitations of ad hoc linear and quadratic multi-level regression models for physical systems, Discrete Contin. Dyn. Syst., Ser. B 17 (4) (2012) 1333–1363.
[47] A.J. Majda, I. Grooms, New perspectives on superparameterization for geophysical turbulence, J. Comput. Phys. 271 (2014) 60–77.
[48] D.S. Wilks, Effects of stochastic parametrizations in the Lorenz'96 system, Q. J. R. Meteorol. Soc. 131 (606) (2005) 389–407.
[49] M. Branicki, A.J. Majda, Dynamic stochastic superresolution of sparsely observed turbulent systems, J. Comput. Phys. 241 (2013) 333–363.
[50] J.L. Anderson, Localization and sampling error correction in ensemble Kalman filter data assimilation, Mon. Weather Rev. 140 (7) (2012) 2359–2371.

[51] N. Chen, A.J. Majda, Beating the curse of dimension with accurate statistics for the Fokker–Planck equation in complex turbulent systems, Proc. Natl. Acad. Sci. USA 114 (49) (2017) 12864–12869.
[52] N. Chen, A.J. Majda, A new efficient parameter estimation algorithm for high-dimensional complex nonlinear turbulent dynamical systems with partial observations, J. Comput. Phys. 397 (2019) 108836.
[53] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. USA 113 (15) (2016) 3932–3937.
[54] N.M. Mangan, S.L. Brunton, J.L. Proctor, J.N. Kutz, Inferring biological networks by sparse identification of nonlinear dynamics, IEEE Trans. Mol. Biol. Multi-Sc. Commun. 2 (1) (2016) 52–63.
[55] C. Novara, Sparse identification of nonlinear functions and parametric set membership optimality analysis, IEEE Trans. Autom. Control 57 (12) (2012) 3236–3241.
[56] F. Santosa, W.W. Symes, Linear inversion of band-limited reflection seismograms, SIAM J. Sci. Stat. Comput. 7 (4) (1986) 1307–1330.
[57] R. Tibshirani, Regression shrinkage and selection via the LASSO, J. R. Stat. Soc., Ser. B, Methodol. 58 (1) (1996) 267–288.
[58] N. Chen, A.J. Majda, D. Giannakis, Predicting the cloud patterns of the Madden-Julian oscillation through a low-order nonlinear stochastic model, Geophys. Res. Lett. 41 (15) (2014) 5612–5619.
[59] N. Chen, A.J. Majda, Predicting the real-time multivariate Madden–Julian oscillation index through a low-order nonlinear stochastic model, Mon. Weather Rev. 143 (6) (2015) 2148–2169.
[60] N. Chen, A.J. Majda, Predicting the cloud patterns for the boreal summer intraseasonal oscillation through a low-order stochastic model, Math. Clim. Weather Forecast. 1 (1) (2015) 1–20.
[61] N. Chen, A.J. Majda, C. Sabeerali, R. Ajayamohan, Predicting monsoon intraseasonal precipitation using a low-order nonlinear stochastic model, J. Climate 31 (11) (2018) 4403–4427.
[62] N. Chen, A.J. Majda, Filtering the stochastic skeleton model for the Madden–Julian oscillation, Mon. Weather Rev. 144 (2) (2016) 501–527.
[63] N. Chen, A.J. Majda, X.T. Tong, Information barriers for noisy Lagrangian tracers in filtering random incompressible flows, Nonlinearity 27 (9) (2014) 2133.
[64] N. Chen, A.J. Majda, X.T. Tong, Noisy Lagrangian tracers for filtering random rotating compressible flows, J. Nonlinear Sci. 25 (3) (2015) 451–488.
[65] N. Chen, A.J. Majda, Model error in filtering random compressible flows utilizing noisy Lagrangian tracers, Mon. Weather Rev. 144 (11) (2016) 4037–4061.
[66] S.R. Keating, A.J. Majda, K.S. Smith, New methods for estimating ocean eddy heat transport using satellite altimetry, Mon. Weather Rev. 140 (5) (2012) 1703–1722.
[67] A.J. Majda, D. Qi, T.P. Sapsis, Blended particle filters for large-dimensional chaotic dynamical systems, Proc. Natl. Acad. Sci. USA (2014) 201405675.
[68] R.S. Liptser, A.N. Shiryaev, Statistics of Random Processes II: Applications, vol. 6, Springer Science & Business Media, 2013.
[69] R.E. Kalman, R.S. Bucy, New results in linear filtering and prediction theory, J. Basic Eng. 83 (1) (1961) 95–108.
[70] P.E. Kloeden, E. Platen, Higher-order implicit strong numerical schemes for stochastic differential equations, J. Stat. Phys. 66 (1–2) (1992) 283–314.
[71] C.W. Gardiner, Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences, Springer Series in Synergetics, vol. 13, 2004.
[72] R. Sundberg, Maximum likelihood theory for incomplete data from an exponential family, Scand. J. Stat. (1974) 49–58.
[73] R. Sundberg, An iterative method for solution of the likelihood equations for incomplete data from exponential families, Commun. Stat., Simul. Comput. 5 (1) (1976) 55–64.
[74] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[75] G. Brassard, P. Bratley, Fundamentals of Algorithmics, vol. 524, Prentice Hall, Englewood Cliffs, 1996.
[76] M. Sorokina, S. Sygletos, S. Turitsyn, Sparse identification for nonlinear optical communication systems: SINO method, Opt. Express 24 (26) (2016) 30433–30443.
[77] Y. Chen, Y. Gu, A.O. Hero, Sparse LMS for system identification, in: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2009, pp. 3125–3128.
[78] D.R. Anderson, Model Based Inference in the Life Sciences: a Primer on Evidence, Springer Science & Business Media, 2007.
[79] Y. Sakamoto, M. Ishiguro, G. Kitagawa, Akaike Information Criterion Statistics, vol. 81, D. Reidel, Dordrecht, the Netherlands, 1986.
[80] G. Schwarz, et al., Estimating the dimension of a model, Ann. Stat. 6 (2) (1978) 461–464.
[81] P. Bühlmann, S. Van De Geer, Statistics for High-Dimensional Data: Methods, Theory and Applications, Springer Science & Business Media, 2011.
[82] A.J. Majda, B. Gershgorin, Quantifying uncertainty in climate change science through empirical information theory, Proc. Natl. Acad. Sci. USA 107 (34) (2010) 14958–14963.
[83] A. Majda, R.V. Abramov, M.J. Grote, Information Theory and Stochastics for Multiscale Nonlinear Systems, vol. 25, American Mathematical Soc., 2005.
[84] R. Kleeman, Information theory and dynamical system predictability, Entropy 13 (3) (2011) 612–649.
[85] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86.
[86] S. Kullback, Letter to the editor: the Kullback-Leibler distance, Am. Stat. 41 (4) (1987) 340–341.
[87] S. Kullback, Statistics and Information Theory, J Wiley Sons, New York, 1997.
[88] G. Branstator, H. Teng, Two limits of initial-value decadal predictability in a CGCM, J. Climate 23 (23) (2010) 6292–6311.
[89] T. DelSole, Predictability and information theory. Part I: measures of predictability, J. Atmos. Sci. 61 (20) (2004) 2425–2440.
[90] T. DelSole, Predictability and information theory. Part ii: imperfect forecasts, J. Atmos. Sci. 62 (9) (2005) 3368–3381.
[91] D. Giannakis, A.J. Majda, Quantifying the predictive skill in long-range forecasting. Part ii: model error in coarse-grained Markov models with application to ocean-circulation regimes, J. Climate 25 (6) (2012) 1814–1826.
[92] R. Kleeman, Measuring dynamical prediction utility using relative entropy, J. Atmos. Sci. 59 (13) (2002) 2057–2072.
[93] H. Teng, G. Branstator, Initial-value predictability of prominent modes of North Pacific subsurface temperature in a CGCM, Clim. Dyn. 36 (9–10) (2011) 1813–1834.
[94] A. Majda, R. Kleeman, D. Cai, et al., A mathematical framework for quantifying predictability through relative entropy, Methods Appl. Anal. 9 (3) (2002) 425–444.
[95] D. Qi, A.J. Majda, Predicting fat-tailed intermittent probability distributions in passive scalar turbulence with imperfect models through empirical information theory, Commun. Math. Sci. 14 (6) (2016) 1687–1722.
[96] E.N. Lorenz, Irregularity: a fundamental property of the atmosphere, Tellus A 36 (2) (1984) 98–110.
[97] E.N. Lorenz, Can chaos and intransitivity lead to interannual variability?, Tellus A 42 (3) (1990) 378–389.
[98] C.B. Muratov, E. Vanden-Eijnden, E. Weinan, Noise can play an organizing role for the recurrent dynamics in excitable media, Proc. Natl. Acad. Sci. USA 104 (3) (2007) 702–707.
[99] C.B. Muratov, E. Vanden-Eijnden, E. Weinan, Self-induced stochastic resonance in excitable systems, Phys. D, Nonlinear Phenom. 210 (3–4) (2005) 227–240.
[100] H. Treutlein, K. Schulten, Noise induced limit cycles of the Bonhoeffer-van der Pol model of neural pulses, Ber. Bunsenges. Phys. Chem. 89 (6) (1985) 710–718.
[101] B. Lindner, L. Schimansky-Geier, Coherence and stochastic resonance in a two-state system, Phys. Rev. E 61 (6) (2000) 6103.

[102] A. Longtin, Stochastic resonance in neuron models, J. Stat. Phys. 70 (1–2) (1993) 309–327.
[103] K. Wiesenfeld, D. Pierson, E. Pantazelou, C. Dames, F. Moss, Stochastic resonance on a circle, Phys. Rev. Lett. 72 (14) (1994) 2125.
[104] A. Neiman, L. Schimansky-Geier, A. Cornell-Bell, F. Moss, Noise-enhanced phase synchronization in excitable media, Phys. Rev. Lett. 83 (23) (1999) 4896.
[105] H. Hempel, L. Schimansky-Geier, J. Garcia-Ojalvo, Noise-sustained pulsating patterns and global oscillations in subexcitable media, Phys. Rev. Lett. 82 (18) (1999) 3713.
[106] B. Hu, C. Zhou, Phase synchronization in coupled nonidentical excitable systems and array-enhanced coherence resonance, Phys. Rev. E 61 (2) (2000) R1001.
[107] P. Jung, A. Cornell-Bell, K.S. Madden, F. Moss, Noise-induced spiral waves in astrocyte syncytia show evidence of self-organized criticality, J. Neurophysiol. 79 (2) (1998) 1098–1101.
[108] N. Chen, A.J. Majda, X.T. Tong, Spatial localization for nonlinear dynamical stochastic models for excitable media, preprint, arXiv:1901.07318.
[109] W.K.-M. Lau, D.E. Waliser, Intraseasonal Variability in the Atmosphere-Ocean Climate System, Springer Science & Business Media, 2011.
[110] J. Berner, U. Achatz, L. Batte, L. Bengtsson, A.d.l. Cámara, H.M. Christensen, M. Colangeli, D.R. Coleman, D. Crommelin, S.I. Dolaptchiev, et al., Stochastic parameterization: toward a new view of weather and climate models, Bull. Am. Meteorol. Soc. 98 (3) (2017) 565–588.
[111] Q. Deng, B. Khouider, A.J. Majda, The MJO in a coarse-resolution GCM with a stochastic multicloud parameterization, J. Atmos. Sci. 72 (1) (2015) 55–74.
[112] R. Plant, G.C. Craig, A stochastic parameterization for deep convection based on equilibrium statistics, J. Atmos. Sci. 65 (1) (2008) 87–105.
[113] D. Olbers, A gallery of simple models from climate physics, in: Stochastic Climate Models, Springer, 2001, pp. 3–63.
[114] J.G. Charney, J.G. DeVore, Multiple flow equilibria in the atmosphere and blocking, J. Atmos. Sci. 36 (7) (1979) 1205–1216.
[115] A.J. Majda, B. Gershgorin, Improving model fidelity and sensitivity for complex systems through empirical information theory, Proc. Natl. Acad. Sci. USA 108 (25) (2011) 10044–10049.
[116] J. Shukla, T. DelSole, M. Fennessy, J. Kinter, D. Paolino, Climate model fidelity and projections of climate change, Geophys. Res. Lett. 33 (7) (2006).
[117] E.N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. 20 (2) (1963) 130–141.
[118] C. Sparrow, The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors, vol. 41, Springer Science & Business Media, 2012.
[119] H. Haken, Analogy between higher instabilities in fluids and lasers, Phys. Lett. A 53 (1) (1975) 77–78.
[120] E. Knobloch, Chaos in the segmented disc dynamo, Phys. Lett. A 82 (9) (1981) 439–440.
[121] M. Gorman, P. Widmann, K. Robbins, Nonlinear dynamics of a convection loop: a quantitative comparison of experiment with theory, Phys. D, Nonlinear Phenom. 19 (2) (1986) 255–267.
[122] N. Hemati, Strange attractors in brushless DC motors, IEEE Trans. Circuits Syst. I, Fundam. Theory Appl. 41 (1) (1994) 40–45.
[123] K.M. Cuomo, A.V. Oppenheim, Circuit implementation of synchronized chaos with applications to communications, Phys. Rev. Lett. 71 (1) (1993) 65.
[124] D. Poland, Cooperative catalysis and chemical chaos: a chemical model for the Lorenz equations, Phys. D, Nonlinear Phenom. 65 (1–2) (1993) 86–99.
[125] S.I. Tzenov, Strange attractors characterizing the osmotic instability, preprint, arXiv:1406.0979.