Supervised learning from noisy observations: Combining machine-learning techniques with data assimilation

Georg A. Gottwald [a],*, Sebastian Reich [b]

[a] *School of Mathematics and Statistics, University of Sydney, NSW 2006, Australia*
[b] *Institute of Mathematics, University of Potsdam, Germany*

ARTICLE INFO

ABSTRACT

Data-driven prediction and physics-agnostic machine-learning methods have attracted increased interest in recent years achieving forecast horizons going well beyond those to be expected for chaotic dynamical systems. In a separate strand of research data-assimilation has been successfully used to optimally combine forecast models and their inherent uncertainty with incoming noisy observations. The key idea in our work here is to achieve increased forecast capabilities by judiciously combining machine-learning algorithms and data assimilation. We combine the physics-agnostic data-driven approach of random feature maps as a forecast model within an ensemble Kalman filter data assimilation procedure. The machine-learning model is learned sequentially by incorporating incoming noisy observations. We show that the obtained forecast model has remarkably good forecast skill while being computationally cheap once trained. Going beyond the task of forecasting, we show that our method can be used to generate reliable ensembles for probabilistic forecasting as well as to learn effective model closure in multi-scale systems.

1. Introduction

Designing computationally cheap models able to predict a system's state in the future is of utmost importance across disciplines in science and engineering. Often modellers face the situation when the underlying model is unknown and has to be inferred from observational data. There exists a plethora of data-driven approaches but most face the problem that their forecast skill is severely impeded by inevitable contamination of observational noise. This work proposes a method that produces cheap surrogate models from noisy observations and provides remarkably good forecast horizons. Data-driven physics-agnostic modelling has become particularly attractive for modelling high-dimensional complex systems where the detailed evolutionary laws are either unknown or are too complex to be resolved numerically, and provide a desirable cost-effective alternative to the numerical simulation of high-dimensional possibly stiff dynamical systems (see, for example, [1–4]). Designing such surrogate dynamical models can be formulated abstractly as a problem of function approximation under supervision: mapping a current state to a new state at a later point in time, using information of a given training data set. Being able to approximate this function with satisfactory accuracy then allows for forecasting by mapping unseen data. The concept of supervised learning can further be exploited to find computationally tractable reduced models for a subset of resolved variables. This problem arises in the context of multi-scale systems where one is typically interested in the dynamics of the slow resolved variables. Reducing a stiff potentially high-dimensional multi-scale system to an effective evolution equation for the slow variables only has obvious computational advantages, reducing the dimensionality and allowing for a larger time step in the numerical discretisation. In this context the aim of machine learning is to learn the so called closure term which parametrises the effect of the fast unresolved degrees of freedom on the resolved slow dynamics. Ideally one aims at determining the closure term when only observations of the resolved variables are available.

A particularly simple and computationally cheap machine learning technique involves random feature maps [5]. In the same spirit as expressing a function as a linear combination of basis functions in some Hilbert space such as for example Fourier or Chebychev basis functions, here functions are expressed as linear combinations of functions $\sigma(\mathbf{w}^{\mathrm{T}}\mathbf{u} + b)$, the so called *random features*, where $\mathbf{w}$ and b are randomly drawn. It was shown that linear combinations of random feature maps are able to approximate continuous functions arbitrarily close, a property known as the universal approximation property [6–8]. We will use the special choice $\sigma(z) = \tanh(z)$ in this paper. The task of training consists of learning the coefficients of the linear combination which can be achieved efficiently by linear ridge regression;

* Corresponding author.
E-mail addresses: georg.gottwald@sydney.edu.au (G.A. Gottwald), sebastian.reich@uni-potsdam.de (S. Reich).

see [9–12] for recent accounts. In the context of dynamical systems the framework of random feature maps was extended to include internal degrees of freedom with their own dynamics in so called echo-state networks [13–16]. Random feature maps and their extensions have been successfully used when the training data set is noise-free, e.g. when the data originate from precise measurements of the model states. In the case when the training data set instead consists of noisy state observations, we shall see that the standard machine learning approach of linear regression becomes suboptimal and provides a mapping that is less suitable for forecasting unseen data.

Incorporating noisy observations to estimate the state of a dynamical system is a problem addressed by data assimilation (DA) or filtering [17–19]. In DA incoming noisy observations are optimally incorporated to increase the predictability of a given forecast model. DA can be used to estimate the state variables as well as unknown parameters of the model. DA now constitutes a standard tool in science and engineering and is used, for example, in numerical weather forecasting where it is the main driver for the increased predictability enjoyed over the past decades. DA has already been used successfully in data-driven modelling where the forecast model is constructed using Takens' embedding theorem and phase space reconstruction [20]. More recently, data assimilation techniques have been combined with techniques from machine learning in order to perform combined state and parameter estimation. We mention in particular the work of [21–25] on smoothing techniques and the recent work of [26] on sequential learning of both state and parameters. Monte Carlo and optimisation based methods for combined state and parameter estimation have already been investigated extensively in the computational statistics community; see [27] for a review. However, most of these techniques are not applicable to the high-dimensional inference problems that arise from semi-parametric representations of the unknown dynamical propagator maps.

Here we shall combine the random feature map architecture within a sequential data assimilation procedure. The idea is to learn the coefficients of the linear combination of the random feature map approximation sequentially by updating them in time with incoming observations rather than performing linear regression on the entire training data set. The forecast model within the data assimilation procedure is given by the random feature map model itself. It should be noted, however, that the resulting combined state-parameter estimation problem is no longer linear. Within such a setting, an attractive DA framework is provided by ensemble Kalman filters (EnKF) where the statistical information needed for the Kalman filter is provided through a Monte Carlo approximation of an ensemble of model forecasts [17–19]. Such ensemble filters were shown to perform very well for nonlinear dynamical systems unlike the classical Kalman filter. We coin our methodology with the acronym RAFDA, standing for RAndom Feature maps and Data Assimilation. We shall consider three prototypical dynamical systems to numerically investigate RAFDA and to illustrate how RAFDA is able to extend the good forecast skills of classical random feature maps to the case of noisy training data. We consider the Lorenz-63 model, the Kuramoto–Sivashinsky equation and the multi-scale Lorenz-96 system to show how RAFDA achieves the goals we set out above. Besides an improved forecast skill of single trajectories compared to classical random feature maps, RAFDA naturally extends to probabilistic forecasting when used in an ensemble setting. We will see that our method generates reliable ensembles for which each ensemble member has equal probability of being closest to the truth. In these applications the aim is to provide a cheap surrogate model which is learned from noisy training data *without* any information about the underlying dynamical system. In the context of multi-scale systems, however, scientists often know the form of the

physical model for the resolved slow scales but lack detailed information about the closure term driving the physical model. The issue of such subgrid-scale parametrisations has recently been addressed anew using machine learning techniques; see for example [28–33]. For the Lorenz-96 multi-scale system we show that given noisy observations of the slow variables RAFDA can determine the closure term and can be used to perform subgrid-scale parametrisation.

The paper is organised as follows. In Section 2 we develop our RAFDA methodology. In Section 3 we present applications to the Lorenz-63 model, where we test in particular for the dependency of the forecast skill on the length of the training data set, the noise level and the reservoir dimension. We further show that ensembles generated through the combined random feature map and data assimilation procedure provide reliable ensembles to be used in probabilistic ensemble forecasting. We further consider the Kuramoto–Sivashinsky equation as a paradigmatic model for spatio-temporal chaos in partial differential equations in Section 4. In Section 5 we consider the multi-scale Lorenz-96 model and show that RAFDA can be used to learn closure models for the effective slow dynamics. We conclude in Section 6 with a discussion and an outlook.

2. Computational methods

Consider a D-dimensional dynamical system

$$\dot{\boldsymbol{u}} = \mathcal{F}(\boldsymbol{u}), \tag{1}$$

which is observed at discrete times $t_n = n\Delta t$ of interval length $\Delta t > 0$, $n = 0, \ldots, N$, to yield a noisy time series of vector-valued observations

$$\boldsymbol{u}_n^{\mathrm{o}} = \boldsymbol{u}_n + \Gamma^{1/2} \boldsymbol{\eta}_n \tag{2}$$

with $\boldsymbol{u}_n^{\mathrm{o}} \in \mathbb{R}^D$, $\boldsymbol{u}_n = \boldsymbol{u}(t_n)$, measurement error covariance matrix $\Gamma \in \mathbb{R}^{D \times D}$ and independent and normally distributed noise $\boldsymbol{\eta}_n \in \mathbb{R}^D$, that is, $\boldsymbol{\eta}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We will typically work with a measurement error covariance matrix

$$\Gamma = \eta \mathbf{I} \tag{3}$$

with scalar variance parameter $\eta \geq 0$. The special case $\eta = 0$ corresponds to exact state observations. In this paper, we are primarily interested in noisy state observations, that is, in $\eta > 0$.

For the purpose of data-driven modelling and machine learning it is instructive to view the evolution of the time-dependent model state $\boldsymbol{u}(t)$ in the time interval Δt as a propagator map

$$\boldsymbol{u}_{n+1} = \Psi_{\Delta t}(\boldsymbol{u}_n). \tag{4}$$

The aim of data-driven modelling is then to find an approximation of this map and construct a surrogate model

$$\hat{\boldsymbol{u}}_{n+1} = \Psi_S(\hat{\boldsymbol{u}}_n) \tag{5}$$

with $\hat{\boldsymbol{u}}_0 = \boldsymbol{u}_0$, which is to be learned from the observational data (2). The observational data may come as outputs from a known model or are given by actual observations without any knowledge of the underlying model. In the case when the model is known a surrogate model may still have computational advantages, in particular for high-dimensional models and/or for stiff multi-scale models, where solving the full model may require prohibitively small step-sizes $\delta t \ll \Delta t$. One can also encounter the case when the underlying dynamics (1) is only partially known. This situation arises typically in multi-scale systems, where $\mathbf{u}(t)$ characterises the resolved (slow) degrees of freedom and the analytic form of $\mathcal{F}$ is known but the effect of the unresolved dynamics on the resolved ones has to be inferred from data.

Independent of whether the underlying model (1) is known or not, one ends up with a combined problem of having to estimate

the states $\boldsymbol{u}_n$ at times $t_n = n\Delta t$ as well as the functional form of the propagator Ψ_S from the noisy data $\boldsymbol{u}_n^o$, $n = 0, \ldots, N$. This problem has been well studied in the literature, both within parametric and non-parametric settings, using different approximation tools such as radial basis functions (RBF) and reproducing kernel Hilbert spaces (RKHS). In this paper, we focus on a particular class of shallow neural networks and the RKHS induced by their random feature maps. See, for example, [10–12] for recent theoretical results on such approximations. In particular, the attractiveness of this RKHS, as compared to more complex machine learning architectures such as those considered for example in [34], is its easy embedding into sequential data assimilation via the ensemble Kalman filter [35], as we will demonstrate in this paper.

We now provide a short derivation of the random feature approximation from a RKHS and Gaussian process perspective [36,37]. The starting point is a kernel function $k(\boldsymbol{u}, \boldsymbol{v})$, which we define to be

$$k(\boldsymbol{u}, \boldsymbol{v}) = \int_{\mathbb{R}^{D+1}} \tanh(\boldsymbol{w}_{\mathrm{in}}^{\mathrm{T}}\boldsymbol{u} + b_{\mathrm{in}}) \tanh(\boldsymbol{w}_{\mathrm{in}}^{\mathrm{T}}\boldsymbol{v} + b_{\mathrm{in}}) \, p(\boldsymbol{w}_{\mathrm{in}}) p(b_{\mathrm{in}}) \, \mathrm{d}\boldsymbol{w}_{\mathrm{in}} \mathrm{d}b_{\mathrm{in}} \tag{6}$$

with $p(\boldsymbol{w}_{\mathrm{in}})$ some multivariate distribution of dimension D and some univariate distribution $p(b_{\mathrm{in}})$. We denote the associated reproducing kernel Hilbert space by $\mathcal{H}_k$. Monte Carlo integration with D_r terms is used to approximate the integral in (6) to obtain

$$\widehat{k}(\boldsymbol{u}, \boldsymbol{v}) = \frac{1}{D_r} \sum_{k=1}^{D_r} \tanh(\boldsymbol{w}_{\mathrm{in},k}^{\mathrm{T}}\boldsymbol{u} + b_{\mathrm{in},k}) \tanh(\boldsymbol{w}_{\mathrm{in},k}^{\mathrm{T}}\boldsymbol{v} + b_{\mathrm{in},k}) \tag{7}$$

with $\boldsymbol{w}_{\mathrm{in},k} \sim p(\boldsymbol{w}_{\mathrm{in}})$ and $b_{\mathrm{in},k} \sim p(b_{\mathrm{in}})$ independently distributed. Using the approximation (7) as the covariance function of a Gaussian process leads to the following random feature representation

$$\Psi_S(\boldsymbol{u}) = \sqrt{\frac{1}{D_r}} \sum_{k=1}^{D_r} w_k \tanh(\boldsymbol{w}_{\mathrm{in},k}^{\mathrm{T}}\boldsymbol{u} + b_{\mathrm{in},k}) \tag{8}$$

of realisations from the Gaussian process $\mathcal{G}(0, \widehat{k})$ with the w_k's independently and standard Gaussian distributed. Note that

$$\widehat{k}(\boldsymbol{u}, \boldsymbol{v}) = \mathbb{E}[\Psi_S(\boldsymbol{u})\Psi_S(\boldsymbol{v})], \tag{9}$$

where the expectation is taken over the random weights w_k in (8). The Gaussian process $\mathcal{G}(0, \widehat{k})$ and its random feature realisations (8) provide a prior for the estimation of the desired propagator $\Psi_S(\boldsymbol{u})$ in (5) in case of $D = 1$.

Our random feature approximation to (5) for $D \geq 1$ proceeds now as follows: We define D_r-dimensional feature maps, which we write in vector form as

$$\boldsymbol{\phi}(\boldsymbol{u}) = \tanh(\boldsymbol{W}_{\mathrm{in}}\boldsymbol{u} + \boldsymbol{b}_{\mathrm{in}}) \in \mathbb{R}^{D_r} \tag{10}$$

using a weight matrix

$$\boldsymbol{W}_{\mathrm{in}} = (\boldsymbol{w}_{\mathrm{in},1}, \ldots, \boldsymbol{w}_{\mathrm{in},D_r})^{\mathrm{T}} \in \mathbb{R}^{D_r \times D}$$

and a bias

$$\boldsymbol{b}_{\mathrm{in}} = (b_{\mathrm{in},1}, \ldots, b_{\mathrm{in},D_r})^{\mathrm{T}} \in \mathbb{R}^{D_r \times 1}.$$

The weight matrix and the bias are chosen randomly and independently of the observed $\boldsymbol{u}_n^o$, $n = 0, \ldots, N$, according to the distributions $p(\boldsymbol{w}_{\mathrm{in}})$ and $p(b_{\mathrm{in}})$, respectively. It is pertinent to mention that here the randomly drawn hyper-parameters $\boldsymbol{W}_{\mathrm{in}}$ and $\boldsymbol{b}_{\mathrm{in}}$ are kept fixed once drawn and are not learned.

The surrogate propagator (4) is then defined by

$$\Psi_S(\boldsymbol{u}) = \boldsymbol{W}\boldsymbol{\phi}(\boldsymbol{u}), \tag{11}$$

where $\boldsymbol{W} \in \mathbb{R}^{D \times D_r}$ is a matrix of coefficients which will be learned from the data $\boldsymbol{u}_n^o$, $n = 0, \ldots, N$. The case $D = 1$ corresponds to

$$\boldsymbol{W} = \left(\frac{w_1}{\sqrt{D_r}}, \ldots, \frac{w_{D_r}}{\sqrt{D_r}} \right) \in \mathbb{R}^{1 \times D_r} \tag{12}$$

in (8). The fact that the coefficients $\boldsymbol{W}$ appear linearly in (11) allows for learning by simple linear regression and makes random feature maps a computationally very attractive network architecture. Despite its simplicity, it was shown that random feature maps and their associated RKHS enjoy the universal approximation property [6–8], ensuring that their associated RKHS is dense in the space of continuous functions. Their success in practical applications, however, depends crucially on an appropriate choice of the two random sets of parameters $\boldsymbol{W}_{\mathrm{in}}$ and $\boldsymbol{b}_{\mathrm{in}}$, that is, the distributions $p(\boldsymbol{w}_{\mathrm{in}})$ and $p(b_{\mathrm{in}})$ in (6). It is also permissible to consider different Gaussian prior distributions for the weights $\boldsymbol{W}$ leading to different Gaussian process priors. We will comment on these aspects in more detail later.

The Monte Carlo approximation perspective on the random feature maps (11) suggests that any function $\Psi_{\Delta t} \in \mathcal{H}_k$ can be approximated by a finite random feature representation (11) with an error rate of $\mathcal{O}(1/\sqrt{D_r})$ which does not contain terms exponentially increasing with the dimension D of the underlying dynamical system (see [38] for a precise statement). The rate constant, however, may be large depending on the underlying dynamical system and its dimension.

In the next two subsection, we discuss how to employ random feature maps to estimate (11) from data. We first describe the standard linear regression approach which assumes a sequence of exact state observations $\boldsymbol{u}_n^o$, $n \geq 0$. We then present our sequential data assimilation approach to the simultaneous estimation of the model states $\boldsymbol{u}_n$ and the model parameters $\boldsymbol{W}$.

Remark. Recently there has been a lot of interest in echo-state networks and reservoir computing [14–16] which involve internal reservoir dynamics. For the examples considered here we find that this additional complexity is not necessary to achieve good forecasting skill. Random feature maps have recently also been extended to solution maps of partial differential equations; see, for example, [39]. All these contributions assume exact state observations and ignore measurement errors.

2.1. Zero measurement noise: Linear regression

If the measurement noise in (2) is zero, that is $\boldsymbol{u}_n = \boldsymbol{u}_n^o$, the external weight matrix $\boldsymbol{W} \in \mathbb{R}^{D \times D_r}$, which maps the random features to the state variable at the next time step, can be determined via linear ridge regression (LR) over a training data set of length N [36,38,39]. More precisely, we seek the minimiser of

$$\mathcal{L}(\boldsymbol{W}) = \frac{1}{2}|\boldsymbol{W}\boldsymbol{R} - \boldsymbol{U}^o|_{\mathrm{F}}^2 + \frac{\beta}{2}|\boldsymbol{W}|_{\mathrm{F}}^2, \tag{13}$$

where $\|\boldsymbol{A}\|_F$ denotes the Frobenius norm of a matrix $\boldsymbol{A}$, $\boldsymbol{U}^o \in \mathbb{R}^{D \times N}$ is the matrix with columns $\boldsymbol{u}_n^o$, $n = 1, \ldots, N$, and $\boldsymbol{R} \in \mathbb{R}^{D_r \times N}$ consists of columns

$$\boldsymbol{\phi}_n = \boldsymbol{\phi}(\boldsymbol{u}_{n-1}^o), \tag{14}$$

$n = 1, \ldots, N$ with fixed draws of the random internal parameters $(\boldsymbol{W}_{\mathrm{in}}, \boldsymbol{b}_{\mathrm{in}})$ and $\boldsymbol{\phi}(\boldsymbol{u})$ given by (10). The parameter $\beta > 0$ is used for regularisation. The minimiser of (13), which we denote by $\boldsymbol{W}_{\mathrm{LR}}$, is explicitly given by

$$\boldsymbol{W}_{\mathrm{LR}} = \boldsymbol{U}^o \boldsymbol{R}^{\mathrm{T}} \left(\boldsymbol{R}\boldsymbol{R}^{\mathrm{T}} + \beta \boldsymbol{I} \right)^{-1}. \tag{15}$$

We remark that if β is chosen too small, the inverse may not be defined and has to be replaced by the pseudo-inverse. Algorithm 1 below provides a pseudocode outlining the relevant input parameters.

Bounds on the empirical risk

$$\mathcal{R}_N(\boldsymbol{W}_{\mathrm{LR}}) = \frac{1}{2N}\|\boldsymbol{W}_{\mathrm{LR}}\boldsymbol{R} - \boldsymbol{U}^{\mathrm{o}}\|_{\mathrm{F}}^2$$

and its generalisation error have been derived in [38] for a regularised cost functional slightly different from (13). More precisely, under some additional assumptions on the kernel function $k(\boldsymbol{u}, \boldsymbol{v})$, the authors obtain a rate of $\mathcal{O}(1/D_r + 1/\sqrt{N})$ provided the regularisation parameter is chosen appropriately.

From a Gaussian process perspective, the linear ridge regression estimator (15) can be interpreted as the MAP estimator to the following Bayesian inference problem [36]: Let us assume $D = 1$ for simplicity and that the prior distribution for the desired propagator map is provided by the Gaussian process $\Psi_S(u) \sim \mathcal{G}(0, \widehat{k})$. Then the negative log-likelihood function of the data is of the form

$$l(\boldsymbol{U}^{\mathrm{o}}|\Psi_S) = \frac{D_r}{2\beta}\sum_{n=1}^{N}\|u_n - \Psi_S(u_{n-1})\|^2,$$

which corresponds to an additive Gaussian model error ξ_n in (5), that is,

$$\hat{u}_{n+1} = \Psi_S(\hat{u}_n) + \xi_n, \tag{16}$$

of mean zero and variance β/D_r. Setting $\beta = D_r$ in (13) corresponds to the standard error model as expected from (12) since $\widehat{k}(u_{n-1}^{\mathrm{o}}, u_{m-1}^{\mathrm{o}}) = D_r^{-1}\boldsymbol{\phi}_n^{\mathrm{T}}\boldsymbol{\phi}_m$. As in many machine learning applications, the data $\boldsymbol{U}^{\mathrm{o}} \in \mathbb{R}^{1\times N}$ is assumed to be noise-free. We shall explore in Sections 3–5 in how far linear ridge regression can be used to construct a forecast model trained on noisy observations $\boldsymbol{U}^{\mathrm{o}}$ and zero model errors, that is $\xi_n \equiv 0$ in (16), instead.

Finding an appropriate weight matrix $\boldsymbol{W}$ in (11) has also been investigated from the perspective of data-driven approximations of the underlying Koopman operator; the so called extended dynamic mode decomposition (EDMD) [40]. The key idea is to first use the complete data set $\boldsymbol{R}$ of some observed user-specified feature maps $\boldsymbol{\phi}(\boldsymbol{u}_n^{\mathrm{o}})$, $n = 0, \ldots, N$, which forms the so called *dictionary*, to construct an approximation of the Koopman operator and to then use its eigenfunctions to approximate the identity map, which in turn implies a computable approximation to the propagator map (5) as a linear combination of the features maps, that is, in the form of (11). This approach is computationally quite involved as it requires the solution of high-dimensional eigenvalue problems. The problem of extracting Koopman operators from noisy data $\boldsymbol{U}^o$ has, for example, been addressed in [41].

We finally mention that the quadratic penalty term in (13) can be replaced by a sparsity enforcing penalty as suggested in [42] and extended to noisy state observations in [43]. It is however not obvious how to extend this, so called, *sparse identification of nonlinear dynamics* (SINDy) approach to an online learning procedure within a Bayesian inference framework.

2.2. Non-zero measurement noise: RAFDA

If the observations are contaminated by noise, we propose here to estimate the weight matrix $\boldsymbol{W}$ recursively using sequential DA [17–19]. Whereas the non-recursive estimation described in the previous section using linear ridge regression only utilises the information contained in the observations, we aim here at optimally estimating the weight matrix using both the observations as well as the underlying dynamical surrogate model (5). In particular, we employ a combined state and parameter estimation

---

Algorithm 1: Linear Regression (LR)

input data : time series $\boldsymbol{u}_n^{\mathrm{o}}$, $n = 1, \ldots, N$

parameters: regularisation parameter β, reservoir dimension D_r, internal parameters $\boldsymbol{W}_{\mathrm{in}} \in \mathbb{R}^{D_r \times D}$, $\boldsymbol{b}_{\mathrm{in}} \in \mathbb{R}^{D_r}$

perform the following:

construct : observation matrix $\boldsymbol{U}^{\mathrm{o}} = \left[\boldsymbol{u}_1^{\mathrm{o}}, \boldsymbol{u}_2^{\mathrm{o}}, \cdots, \boldsymbol{u}_N^{\mathrm{o}}\right] \in \mathbb{R}^{D\times N}$

random features $\boldsymbol{\phi}_n = \tanh(\boldsymbol{W}_{\mathrm{in}}\boldsymbol{u}_{n-1}^{\mathrm{o}} + \boldsymbol{b}_{\mathrm{in}}) \in \mathbb{R}^{D_r}$

feature matrix $\boldsymbol{R} = \left[\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots, \boldsymbol{\phi}_N\right] \in \mathbb{R}^{D_r \times N}$

output : $\boldsymbol{W}_{\mathrm{LR}} = \boldsymbol{U}^{\mathrm{o}}\boldsymbol{R}^{\mathrm{T}}\left(\boldsymbol{R}\boldsymbol{R}^{\mathrm{T}} + \beta\mathbf{I}\right)^{-1}$

---

via state augmentation [17,19]. More precisely, we formulate the forecast model for constant parameters $\boldsymbol{W}$ as

$$\boldsymbol{u}_{n+1}^{\mathrm{f}} = \boldsymbol{W}_n^{\mathrm{a}}\,\boldsymbol{\phi}(\boldsymbol{u}_n^{\mathrm{a}}) \tag{17a}$$

$$\boldsymbol{W}_{n+1}^{\mathrm{f}} = \boldsymbol{W}_n^{\mathrm{a}}, \tag{17b}$$

where the superscript f denotes the forecast and the superscript a denotes the analysis defined below.

Furthermore, the model states $\boldsymbol{u}_n^{\mathrm{a}}$, $\boldsymbol{W}_n^{\mathrm{a}}$ and $\boldsymbol{u}_{n+1}^{\mathrm{f}}$, $\boldsymbol{W}_{n+1}^{\mathrm{f}}$, respectively, are now treated as random variables. To formulate the Kalman filter analysis step, we unravel the weight matrix $\boldsymbol{W} \in \mathbb{R}^{D\times D_r}$ into a parameter vector $\boldsymbol{w} \in \mathbb{R}^{DD_r}$ with $w_{1:D_r} = W_{11}, \ldots, W_{1D_r}$, $w_{D_r+1:2D_r} = W_{21}, \ldots, W_{2D_r}$ and so forth. Concatenating further we introduce $\boldsymbol{z} = (\boldsymbol{u}^{\mathrm{T}}, \boldsymbol{w}^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{D_z}$ with $D_z = D + DD_r$. While the forecast step (17) leads to an update of the random variable $\boldsymbol{z}_n^{\mathrm{a}}$ into $\boldsymbol{z}_{n+1}^{\mathrm{f}}$, the analysis step for the mean $\bar{\boldsymbol{z}}_{n+1}^{\mathrm{a}}$ is provided by

$$\bar{\boldsymbol{z}}_{n+1}^{\mathrm{a}} = \bar{\boldsymbol{z}}_{n+1}^{\mathrm{f}} - \boldsymbol{K}_{n+1}(\boldsymbol{H}\bar{\boldsymbol{z}}_{n+1}^{\mathrm{f}} - \boldsymbol{u}_{n+1}^{\mathrm{o}}) \tag{18}$$

with the observation matrix $\boldsymbol{H} \in \mathbb{R}^{D\times D_z}$ defined by $\boldsymbol{H}\boldsymbol{z} = \boldsymbol{u}$, i.e., we assume that we observe all state variables $\boldsymbol{u}$. The Kalman gain matrix $\boldsymbol{K}$ is given by

$$\boldsymbol{K}_{n+1} = \boldsymbol{P}_{n+1}^{\mathrm{f}}\boldsymbol{H}^{\mathrm{T}}\left(\boldsymbol{H}\boldsymbol{P}_{n+1}^{\mathrm{f}}\boldsymbol{H}^{\mathrm{T}} + \Gamma\right)^{-1}, \tag{19}$$

where the forecast covariance matrix is given by

$$\boldsymbol{P}_{n+1}^{\mathrm{f}} = \langle\hat{\boldsymbol{z}}_{n+1}^{\mathrm{f}} \otimes \hat{\boldsymbol{z}}_{n+1}^{\mathrm{f}}\rangle = \begin{pmatrix} \langle\hat{\boldsymbol{u}}_{n+1}^{\mathrm{f}} \otimes \hat{\boldsymbol{u}}_{n+1}^{\mathrm{f}}\rangle & \langle\hat{\boldsymbol{u}}_{n+1}^{\mathrm{f}} \otimes \hat{\boldsymbol{w}}_{n+1}^{\mathrm{f}}\rangle \\ \langle\hat{\boldsymbol{w}}_{n+1}^{\mathrm{f}} \otimes \hat{\boldsymbol{u}}_{n+1}^{\mathrm{f}}\rangle & \langle\hat{\boldsymbol{w}}_{n+1}^{\mathrm{f}} \otimes \hat{\boldsymbol{w}}_{n+1}^{\mathrm{f}}\rangle \end{pmatrix}. \tag{20}$$

The angular brackets $\langle f(\boldsymbol{z})\rangle$ denote the expectation value of a function $f(\boldsymbol{z})$ and the hat denotes the perturbation of $\boldsymbol{z}_{n+1}^{\mathrm{f}}$ from its mean $\bar{\boldsymbol{z}}_{n+1}^{\mathrm{f}} = \langle\boldsymbol{z}_{n+1}^{\mathrm{f}}\rangle$, that is,

$$\hat{\boldsymbol{u}}_{n+1}^{\mathrm{f}} = \boldsymbol{u}_{n+1}^{\mathrm{f}} - \bar{\boldsymbol{u}}_{n+1}^{\mathrm{f}}. \tag{21}$$

Since we are only observing the state variables $\boldsymbol{u}$, the Kalman update (18) can be written to explicitly separate the state and parameter variables as

$$\bar{\boldsymbol{u}}_{n+1}^{\mathrm{a}} = \bar{\boldsymbol{u}}_{n+1}^{\mathrm{f}} - \boldsymbol{P}_{\boldsymbol{uu}}^{\mathrm{f}}\left(\boldsymbol{P}_{\boldsymbol{uu}}^{\mathrm{f}} + \Gamma\right)^{-1}\left(\bar{\boldsymbol{u}}_{n+1}^{\mathrm{f}} - \boldsymbol{u}_{n+1}^{\mathrm{o}}\right) \tag{22}$$

$$\bar{\boldsymbol{w}}_{n+1}^{\mathrm{a}} = \bar{\boldsymbol{w}}_{n+1}^{\mathrm{f}} - \boldsymbol{P}_{\boldsymbol{wu}}^{\mathrm{f}}\left(\boldsymbol{P}_{\boldsymbol{uu}}^{\mathrm{f}} + \Gamma\right)^{-1}\left(\bar{\boldsymbol{u}}_{n+1}^{\mathrm{f}} - \boldsymbol{u}_{n+1}^{\mathrm{o}}\right), \tag{23}$$

where $\boldsymbol{P}_{\boldsymbol{uu}}^{\mathrm{f}} = \langle\hat{\boldsymbol{u}}_{n+1}^{\mathrm{f}} \otimes \hat{\boldsymbol{u}}_{n+1}^{\mathrm{f}}\rangle$ and $\boldsymbol{P}_{\boldsymbol{wu}}^{\mathrm{f}} = \langle\hat{\boldsymbol{w}}_{n+1}^{\mathrm{f}} \otimes \hat{\boldsymbol{u}}_{n+1}^{\mathrm{f}}\rangle$.

The Kalman filter is optimal in the sense that $\bar{\boldsymbol{z}}^a$ maximises the likelihood of the observations provided that both the observations and the state variables are Gaussian distributed random variables. Since the forward model (17) is nonlinear in the augmented state variables and the involved random variables cannot be assumed to be Gaussian distributed, the combined forecast and analysis cycle is not well defined and we employ the stochastic EnKF [35, 44] to define a computationally robust Monte Carlo closure. In

particular, define an ensemble of states $\boldsymbol{Z} \in \mathbb{R}^{D_z \times M}$ consisting of M members $\boldsymbol{z}^{(i)} \in \mathbb{R}^{D_z}$, $i = 1, \ldots, M$, that is,

$$\boldsymbol{Z} = \left[\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots, \boldsymbol{z}^{(M)} \right], \tag{24}$$

with empirical mean

$$\bar{\boldsymbol{z}} = \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{z}^{(i)}, \tag{25}$$

and the associated matrix of ensemble deviations

$$\hat{\boldsymbol{Z}} = \left[\boldsymbol{z}^{(1)} - \bar{\boldsymbol{z}}, \boldsymbol{z}^{(2)} - \bar{\boldsymbol{z}}, \ldots, \boldsymbol{z}^{(M)} - \bar{\boldsymbol{z}} \right] \in \mathbb{R}^{D_z \times M}. \tag{26}$$

We define these objects for the forecast and analysis ensembles $\boldsymbol{Z}_n^{\mathrm{f}}$ and $\boldsymbol{Z}_n^{\mathrm{a}}$, respectively, for all $n \geq 0$. Each ensemble member is propagated individually under the forecast step (17). This defines the update from the last analysis ensemble $\boldsymbol{Z}_n^{\mathrm{a}}$ to the next forecast ensemble $\boldsymbol{Z}_{n+1}^{\mathrm{f}}$. The EnKF analysis step is now defined as follows. The ensemble deviation matrix $\hat{\boldsymbol{Z}}_{n+1}^{\mathrm{f}}$ can be used to approximate the forecast covariance matrix (20) as

$$\boldsymbol{P}_{n+1}^{\mathrm{f}} = \frac{1}{M-1} \hat{\boldsymbol{Z}}_{n+1}^{\mathrm{f}} (\hat{\boldsymbol{Z}}_{n+1}^{\mathrm{f}})^{\mathrm{T}} \in \mathbb{R}^{D_z \times D_z}. \tag{27}$$

Upon introducing the matrix $\boldsymbol{U}_{n+1}^{\mathrm{o}} \in \mathbb{R}^{D \times M}$ of perturbed observations

$$\boldsymbol{U}_{n+1}^{\mathrm{o}} = \left[\boldsymbol{u}_{n+1}^{\mathrm{o}} - \Gamma^{1/2} \boldsymbol{\eta}_{n+1}^{(1)}, \boldsymbol{u}_{n+1}^{\mathrm{o}} - \Gamma^{1/2} \boldsymbol{\eta}_{n+1}^{(2)}, \ldots, \boldsymbol{u}_{n+1}^{\mathrm{o}} - \Gamma^{1/2} \boldsymbol{\eta}_{n+1}^{(M)} \right], \tag{28}$$

where the $\boldsymbol{\eta}_{n+1}^{(i)} \in \mathbb{R}^D$, $i = 1, \ldots, M$, are realisations of independent and normally distributed random variables (cf (2)), we obtain the following compact representation of the EnKF update step:

$$\boldsymbol{Z}_{n+1}^{\mathrm{a}} = \boldsymbol{Z}_{n+1}^{\mathrm{f}} - \boldsymbol{P}_{n+1}^{\mathrm{f}} \boldsymbol{H}^{\mathrm{T}} \left(\boldsymbol{H} \boldsymbol{P}_{n+1}^{\mathrm{f}} \boldsymbol{H}^{\mathrm{T}} + \Gamma \right)^{-1} \left(\boldsymbol{H} \boldsymbol{Z}_{n+1}^{\mathrm{f}} - \boldsymbol{U}_{n+1}^{\mathrm{o}} \right). \tag{29}$$

The ensemble forecast step defined by (17), together with the EnKF analysis step (29) constitute our combined RAndom Features maps and Data Assimilation (RAFDA) method. Training is done with a single long trajectory of length N, and the ensemble mean of the weight matrix $\boldsymbol{W}_N^{\mathrm{a}}$ at final training time $t_N = \Delta t N$ is denoted by $\boldsymbol{W}_{\mathrm{RAFDA}}$. We note that random perturbations, such as in (16) can easily be included into the forecast step (17a). Throughout this paper we assume however that the models are deterministic as in (4). In the subsequent two subsections we discuss further algorithmic details of our method.

2.3. Choice of random feature maps

The choice of the random coefficients $\boldsymbol{W}_{\mathrm{in}}$ and $\boldsymbol{b}_{\mathrm{in}}$ in the feature maps (10) and their dimension D_r is crucial for the success of our method. In this study, we choose these entries to be independent and uniformly distributed with

$$(\boldsymbol{W}_{\mathrm{in}})_{ij} \sim \mathcal{U}[-w, w] \qquad \text{and} \qquad (\boldsymbol{b}_{\mathrm{in}})_i \sim \mathcal{U}[-b, b]. \tag{30}$$

The hyper-parameters $w > 0$ and $b > 0$ should be chosen such that the random feature maps (14) evaluated at the observed data points cover their full range of $[-1, 1]^D$ as uniformly as possible, in particularly sampling their nonlinear domain. See the numerical experiment section for more details.

We mention that one could also dynamically adapt the internal parameters $(\boldsymbol{W}_{\mathrm{in}}, \boldsymbol{b}_{\mathrm{in}})$. This would extend our method from random feature maps to two-layer networks, going from RKHS to Barron spaces [45–49]. We leave this option in combination with data assimilation as a topic for further research.

2.4. Further algorithmic details of RAFDA

The required ensemble size M of the standard implementation (29) of an EnKF needs to be of the order $\mathcal{O}(D_z)$, which can become prohibitive for typical reservoir dimensions of $D_r \sim \mathcal{O}(10^3)$ and potentially high-dimensional dynamical systems. Within the EnKF community further approximations such as localisation and inflation have been developed to deal with finite-size effects [19,35,50]. Here we follow the concept of B-localisation [51–53], where the empirical covariance matrix $\boldsymbol{P}_{n+1}^{\mathrm{f}}$ is tempered by a symmetric positive definite localisation matrix $\mathbf{B}$ via the Kronecker product, that is,

$$\widetilde{\boldsymbol{P}}_{n+1}^{\mathrm{f}} = \mathbf{B} \circ \boldsymbol{P}_{n+1}^{\mathrm{f}}. \tag{31}$$

The EnKF update step (29) is then replaced by

$$\boldsymbol{Z}_{n+1}^{\mathrm{a}} = \boldsymbol{Z}_{n+1}^{\mathrm{f}} - \widetilde{\boldsymbol{P}}_{n+1}^{\mathrm{f}} \boldsymbol{H}^{\mathrm{T}} \left(\boldsymbol{H} \widetilde{\boldsymbol{P}}_{n+1}^{\mathrm{f}} \boldsymbol{H}^{\mathrm{T}} + \Gamma \right)^{-1} \left(\boldsymbol{H} \boldsymbol{Z}_{n+1}^{\mathrm{f}} - \boldsymbol{U}_{n+1}^{\mathrm{o}} \right). \tag{32}$$

Localisation implies that certain correlations between variables get reduced or even set to zero if the corresponding entry in $\mathbf{B}$ is less than one or zero, respectively. This allows to control spurious correlations between uncorrelated variables of $\mathcal{O}(1/\sqrt{M})$, caused by the finite ensemble size. In our numerical experiments we employ the following form of localisation. The jth row of the weight matrix $\boldsymbol{W}$ is responsible for updating the jth component of the dynamic state variable $\boldsymbol{u}$ under the forward model (17a). We reflect this property in the approximate covariance matrix $\widetilde{\boldsymbol{P}}_{n+1}^{\mathrm{f}}$ by ignoring all correlations between the jth row of $\boldsymbol{W}$ and all components of the observed $\boldsymbol{u}_{n+1}^{\mathrm{o}}$ except for its jth entry. We still require M to be $\mathcal{O}(D_r)$ to mitigate against possible spurious correlations. The issue of localisation in the context of combined state and parameter estimation has also been addressed recently in [26].

We also employ multiplicative inflation in which the ensemble members $\boldsymbol{z}_{n+1}^{(i)}$ of the forecast ensemble $\boldsymbol{Z}_{n+1}^{\mathrm{f}}$ are replaced by

$$\boldsymbol{z}_{n+1}^{(i)} \rightarrow \bar{\boldsymbol{z}}_{n+1} + \alpha(\boldsymbol{z}_{n+1}^{(i)} - \bar{\boldsymbol{z}}_{n+1}), \tag{33}$$

$i = 1, \ldots, M$, prior to the EnKF data assimilation step (29) with $\alpha > 1$ being the inflation factor [54]. Note that the inflation step maintains the ensemble mean while increasing the covariance matrix $\boldsymbol{P}_{n+1}^{\mathrm{f}}$ of the ensemble.

We finally need to specify the distribution of the extended state variable $\boldsymbol{z}$ at initial time $t = 0$ from which to draw our initial ensemble. We treat the two components $\boldsymbol{u}_0$ and $\boldsymbol{w}_0$ of $\boldsymbol{z}_0$ as independent, and set

$$\boldsymbol{u}_0 \sim \mathcal{N}(\boldsymbol{u}_0^{\mathrm{o}}, \Gamma) \tag{34}$$

and

$$\boldsymbol{w}_0 \sim \mathcal{N}(\boldsymbol{w}_{\mathrm{LR}}, \gamma \mathbf{I}), \tag{35}$$

where $\boldsymbol{w}_{\mathrm{LR}}$ is the vectorial form of the solution $\boldsymbol{W}_{\mathrm{LR}}$ to the ridge regression formulation (13) and $\gamma > 0$ is a parameter. Alternatively, one can also use $\boldsymbol{w}_0 \sim \mathcal{N}(\mathbf{0}, \gamma \mathbf{I})$ if $\boldsymbol{w}_{\mathrm{LR}}$ is unavailable.

We recall that the linear regression formulation (13) ignores measurement errors in the states, that is, assumes $\boldsymbol{u}_0 = \boldsymbol{u}_0^{\mathrm{o}}$. The regularisation term in (13) corresponds to the choice of prior for $\boldsymbol{w}_0$ in (35). However, we re-emphasise that LR and RAFDA solve different estimation problems.

We provide a pseudocode outlining the relevant input parameters in Algorithm 2. In the subsequent sections we illustrate the proposed RAFDA methodology for several dynamical systems, demonstrate its improved forecast skill compared to the linear regression approach (15), and investigate the dependency of RAFDA on the noise strength and the available training size N, as well as

showing the impact on the approximation properties of RAFDA of the various choices a modeller needs to take such as the reservoir dimension D_r and the choice of the random elements $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{b}_{\text{in}}$.

Algorithm 2: Random Feature Map Data Assimilation (RAFDA)

---

input data : time series $\boldsymbol{u}_n^{\text{o}}$, $n = 0, \ldots, N$
parameters: random feature maps: dimension D_r, internal
            parameters $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{D_r \times D}$, $\boldsymbol{b}_{\text{in}} \in \mathbb{R}^{D_r}$
            EnKF: ensemble size M, observational noise
            covariance $\boldsymbol{\Gamma}$, inflation α,
            initial ensemble parameters $(\boldsymbol{w}_{\text{LR}}, \gamma)$

perform the following:
initialising ensemble:
    $\boldsymbol{Z}_0^{\text{a}}$ *with members drawn according to* $\boldsymbol{u}_0 \sim \mathcal{N}(\boldsymbol{u}_0^{\text{o}}, \boldsymbol{\Gamma})$ *and*
$\boldsymbol{w}_0 \sim \mathcal{N}(\boldsymbol{w}_{\text{LR}}, \gamma \boldsymbol{I})$;
for $n = 1 : N$ **do**
    forecast $\boldsymbol{Z}_{n-1}^{\text{a}} \rightarrow \boldsymbol{Z}_n^{\text{f}}$: *each ensemble member is propagated*
    according to
       $\boldsymbol{u}_n^{\text{f}} = \boldsymbol{W}_{n-1}^{\text{a}} \phi(\boldsymbol{u}_{n-1}^{\text{a}})$;
       $\boldsymbol{W}_n^{\text{f}} = \boldsymbol{W}_{n-1}^{\text{a}}$;
    data assimilation analysis update $\boldsymbol{Z}_n^{\text{f}} \rightarrow \boldsymbol{Z}_n^{\text{a}}$:
       inflation & localisation: $\boldsymbol{P}_n^{\text{f}} \leftarrow \mathbf{B} \circ \boldsymbol{P}_n^{\text{f}}$
       $\boldsymbol{Z}_n^{\text{a}} = \boldsymbol{Z}_n^{\text{f}} - \boldsymbol{P}_n^{\text{f}} \boldsymbol{H}^{\text{T}} \left(\boldsymbol{H} \boldsymbol{P}_n^{\text{f}} \boldsymbol{H}^{\text{T}} + \boldsymbol{\Gamma} \right)^{-1} \left(\boldsymbol{H} \boldsymbol{Z}_n^{\text{f}} - \boldsymbol{U}_n^{\text{o}} \right)$;
output   : $\boldsymbol{W}_{\text{RAFDA}} = $ ensemble average of $\boldsymbol{W}_N^{\text{a}}$

---

3. Ordinary differential equations: Lorenz-63 equation

We consider as a first test bed the Lorenz-63 system [55]

$$\dot{y} = 10(y - x)$$
$$\dot{x} = 28x - y - xz$$
$$\dot{z} = -\frac{8}{3}z + xy \tag{36}$$

with $\boldsymbol{u} = (x, y, z)^{\text{T}} \in \mathbb{R}^3$. Observations are taken every $\Delta t = 0.02$ time units and $\boldsymbol{u}_n$ corresponds to the solution at $t_n = n\Delta t$. In all simulations a transient of 40 model time units is discarded to ensure that the dynamics has settled on the attractor.

To assess the propensity for random feature maps with data assimilation and for classical random feature maps with linear regression to be used as a forecast model we test the forecast models of LR

$$\boldsymbol{u}_{n+1} = \boldsymbol{W}_{\text{LR}} \, \boldsymbol{\phi}(\boldsymbol{u}_n) \tag{37}$$

with $\boldsymbol{W}_{\text{LR}}$ given by (15), and of RAFDA

$$\boldsymbol{u}_{n+1} = \boldsymbol{W}_{\text{RAFDA}} \, \boldsymbol{\phi}(\boldsymbol{u}_n), \tag{38}$$

where $\boldsymbol{W}_{\text{RAFDA}}$ is determined via the training data as the final outcome of the iterative DA procedure given by (17) and (29). In both cases the reservoir variables are constructed using (10). For LR we employ a regularisation parameter of $\beta = 4 \times 10^{-5}$ (unless stated otherwise). For RAFDA, unless stated otherwise, we do not use inflation ($\alpha = 0$) and set $D_r = M = 300$ and $N = 4{,}000$. We assume that we know the observational noise strength η.

We test the respective forecast capabilities in a validation data set $\boldsymbol{u}_{\text{valid}}(t_n)$, $n \geq 0$. As a quantitative diagnostic for the forecast skill we record the forecast time τ_f, defined as the largest time such that the relative forecast error $\mathcal{E}(t_n) = \|\boldsymbol{u}_{\text{valid}}(t_n) - \boldsymbol{u}_n\|^2 / \|\boldsymbol{u}_{\text{valid}}(t_n)\|^2 \leq \theta$. We choose here $\theta = 0.05$. This threshold was chosen to correspond with an eye-ball metric that the truth and the surrogate model have diverged.

We measure time in units of the Lyapunov time $t\lambda_{\text{max}}$ with the maximal Lyapunov exponent $\lambda_{\text{max}} = 0.91$. We shall present results for the mean behaviour over realisations, which differ in observations used for the training, in the validation data set, each of which with independently drawn initial conditions, as well as in the random draws of the internal parameters ($\boldsymbol{W}_{\text{in}}$, $\boldsymbol{b}_{\text{in}}$) and of the initial ensembles for RAFDA (unless stated otherwise). The different training and validation data sets are generated by drawing random initial conditions, which then each subsequently evolve into random dynamic states on the attractor after a transient time of 40 model time units. In the following we discuss the effect of the various parameters as well as on the length of the available data and the noise levels on the forecast capabilities of the two models.

3.1. Dependency on the internal parameters

We first address the issue of the choice of the randomly chosen internal parameters $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{b}_{\text{in}}$. The entries of the internal weight matrix and bias are chosen uniformly randomly over intervals $[-w, w]$ in case of $\boldsymbol{W}_{\text{in}}$ and $[-b, b]$ in case of $\boldsymbol{b}_{\text{in}}$ (cf (30)). Fig. 1 shows the mean forecast times τ_f and the associated standard deviations as a function of the two parameters w and b averaged over 50 independent realisations of (30) using the same training data set and the same validation data set $\boldsymbol{u}_{\text{valid}}$ but differing in the random draws ($\boldsymbol{W}_{\text{in}}$, $\boldsymbol{b}_{\text{in}}$). The forecast times are estimated using the ridge linear regression matrix $\boldsymbol{W}_{\text{LR}}$ with $N = 20{,}000$ training data points. Results are shown for training on noiseless and on noisy observations with $\eta = 0.2$. The forecast capabilities clearly depend sensitively on the choice of w and b. In particular, parameter choices leading to excellent forecast times in the noiseless case may not lead to good forecast times in the case of non-zero measurement noise (note the different range in w for the contour plots shown in Fig. 1). The optimal forecast times τ_f for noiseless data is 5.5 whereas it drops to 1.8 when trained on noisy data, illustrating that random feature maps have difficulties when trained on noisy data. Moreover, in the noiseless case parameter combinations (w, b) associated with large forecast times exhibit small variances which renders random feature maps and linear regression insensitive to the particular random draw. In the case of noisy data, however, it is seen that parameter combinations (w, b) corresponding to large forecast times also have larger standard deviations, rendering this method unreliable for noisy data.

To explore what constitutes a good set of internal parameters corresponding to long forecast times we show in Fig. 2 normalised histograms of one of the components of the random feature map ϕ_j (cf (10)). We pick values (w, b) which we identified to lead (on average) to poor forecast times $\tau_f < 0.5$ and which we identified to lead to good forecast times with $\tau_f > 4$ in the case of noiseless observations (cf. Fig. 1). We then choose randomly drawn typical realisations corresponding to each of the (w, b). The histogram is then generated by evaluating ϕ_1 along a time trajectory $\boldsymbol{u}(t)$ of the Lorenz-63 system of length $N = 200{,}000$ sampled every 0.02 time steps, to explore the range of the random feature map ϕ_1 for the Lorenz-63 dynamics. For internal parameter choices corresponding to poor forecast times $\tau_f < 0.5$ the histograms exhibit strongly localised behaviour near ± 1, implying that the tanh-function cannot resolve the whole range of dynamical states of the underlying observations. Internal parameters corresponding to long forecast times on the other hand resolve the whole dynamical range within the nonlinear range of the tanh-function. In the following the internal weight matrix and bias are chosen randomly with $w = 0.005$ and $b = 4$ which yields (on average) large forecast times τ_f for $\eta = 0$ and $\eta = 0.2$ (cf. Fig. 1).
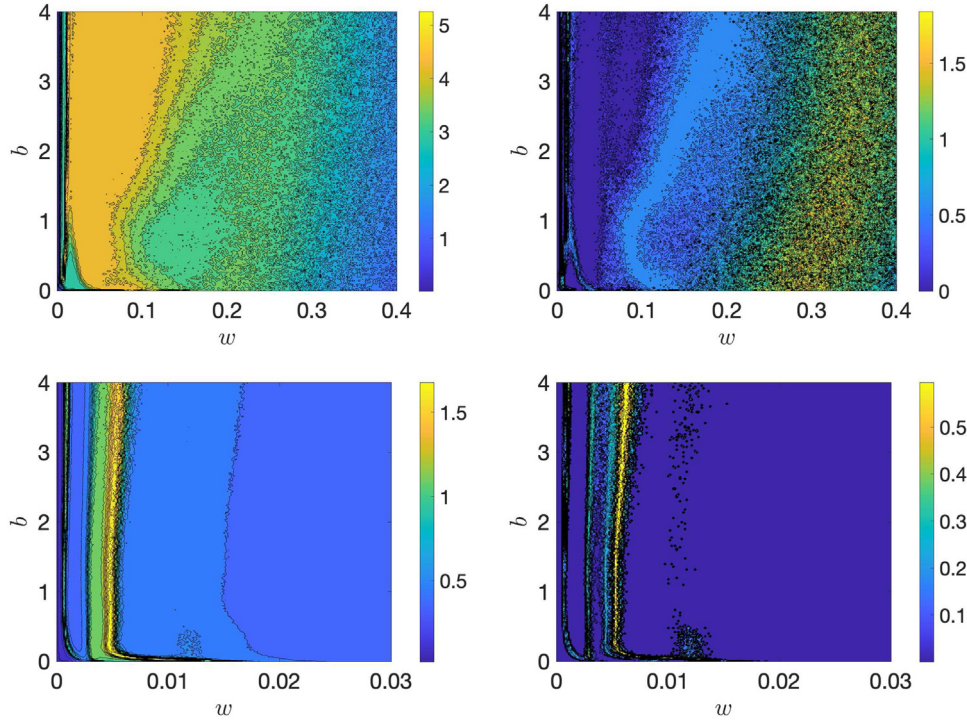
Fig. 1. Contourplot of the forecast time τ_f in units of the Lyapunov time (left) for different choices of the internal parameters, averaged over 50 realisations, and the associated standard deviations (right) computed using LR. Top: noiseless observations with $\eta = 0$. Bottom: noisy observations with $\eta = 0.2$.
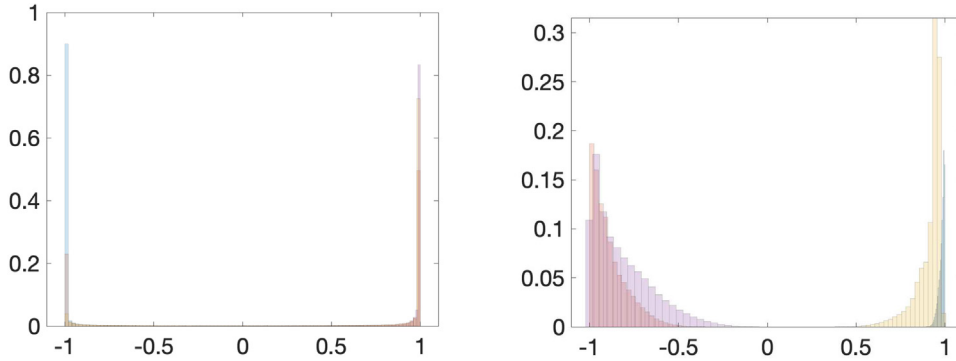
Fig. 2. Normalised empirical histogram of the random feature map (10) ϕ_1 for four choices of internal parameters $(\boldsymbol{W}_{\text{in}}, \boldsymbol{b}_{\text{in}})$, drawn for four separate values of (w, b). Left: for internal parameters corresponding to forecast times $\tau_f < 0.5$. Right: for internal parameters corresponding to forecast times $\tau_f > 6$.

3.2. Dependency of LR on the regularisation parameter β

Fig. 3 shows the mean of the forecast time τ_f obtained from LR for reservoir dimension $D_r = 300$ for different choices of the regularisation parameter β, averaged over 500 realisations, each trained on $N = 4{,}000$ observations. Each realisation uses a different random draw of the internal parameters $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{b}_{\text{in}}$ as well as different training and validation data sets. For values of the regularisation parameter $\log \beta < -16$ the inverse in the LR solution (15) is badly scaled and we instead use the pseudo-inverse (we use the natural logarithm throughout the paper). We observe that when noiseless data are used to train LR, the forecast time plateaus to its maximal value of around $\tau_f = 3.8$ for $\log \beta < -20$, including the case of no regularisation with $\beta = 0$. For noisy observations with $\eta = 0.2$ the mean forecast time τ_f is relatively robust to changes in the regularisation parameter for a wide range of β with $\log \beta \in [-25, -7]$. Within this range

the forecast times τ_f are consistently larger for the training set consisting of noiseless observations. In the following we employ a regularisation parameter of $\beta = 4 \times 10^{-5}$ ($\log \beta \approx -10$).

3.3. Dependency of RAFDA on the initial ensemble

Turning to RAFDA, we now investigate the influence of the choice of the initial ensemble for the parameters on its performance. We consider initial ensembles drawn in an unbiased fashion with $\boldsymbol{w}_0 \sim \mathcal{N}(\boldsymbol{0}, \gamma \boldsymbol{I})$ as well as drawn around the prior provided by LR with $\boldsymbol{w}_0 \sim \mathcal{N}(\boldsymbol{w}_{\text{LR}}, \gamma \boldsymbol{I})$ as in (35). Recall that the small-cap $\boldsymbol{w}$ denotes the vectorial representation of the parameter matrix large-cap $\boldsymbol{W}$ which is learned in RAFDA. The model is trained on $N = 4{,}000$ noisy observations with noise level $\eta = 0.2$ and a reservoir of dimension $D_r = 300$ was used. We chose $M = 300$ ensemble members for RAFDA, and averaged over 500 realisations, each involving independently drawn random
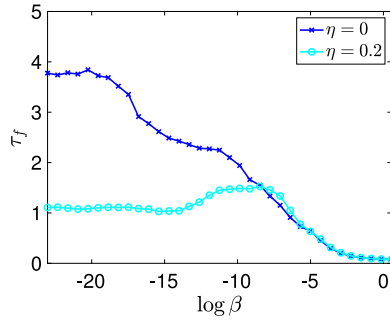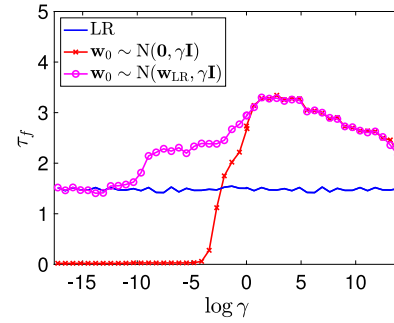
Fig. 3. Mean of the forecast time τ_f in units of the Lyapunov time for varying regularisation parameters β for LR for noiseless ($\eta = 0$) and for noise contaminated observations ($\eta = 0.2$).

Fig. 4. Mean of the forecast time τ_f in units of the Lyapunov time for varying γ for RAFDA for noise contaminated observations with $\eta = 0.2$. Shown are results when the initial ensemble is drawn around the prior provided by LR $\boldsymbol{w}_0 \sim \mathcal{N}(\boldsymbol{w}_{\mathrm{LR}}, \gamma\mathbf{I})$ (35) and for the unbiased choice $\boldsymbol{w}_0 \sim \mathcal{N}(\mathbf{0}, \gamma\mathbf{I})$.

internal parameters $\boldsymbol{W}_{\mathrm{in}}$ and $\boldsymbol{b}_{\mathrm{in}}$, independently drawn initial ensembles for RAFDA as well as different training and validation data sets. Fig. 4 shows the dependency of the mean forecast time on the initial spread of the ensemble γ. At each value of γ choosing the initial ensemble to be centred around the prior provided by LR produces forecast times which are better than or equal to those obtained with an unbiased initial ensemble for the same value of γ. The forecast time vacillates around $\tau_f \approx 1.5$ for LR (the small fluctuations occur due to each realisation having different independent training and validation data sets). For both types of initial RAFDA ensembles there is a wide range of initial ensemble spreads γ for which good forecast times are obtained. When γ is too small and the spread of the initial ensemble in the data assimilation component of RAFDA is too small, EnKF experiences filter divergence and collapses, implying a small error covariance matrix $\boldsymbol{P}^{\mathrm{f}}$. This leads to the filter trusting its own forecast, ignoring the information from incoming observations. The filter then amounts to running the bare forecast model, leading to poor forecast skills with small $\tau_f \to 0$ for $\gamma \to 0$. Centring the initial ensemble around the LR prior $\boldsymbol{w}_{\mathrm{LR}}$ significantly delays this filter collapse with forecast times $\tau_f \approx 2.5$ when the unbiased initial ensemble has already collapsed near $\log\gamma \approx -4$. When further decreasing the spread of the initial parameter ensemble γ with $\log\gamma \lessapprox -10$, the parameters $\boldsymbol{w}$ of RAFDA, when initialised on the prior provided by LR, are not corrected by the incoming observations and RAFDA and LR perform equally well.

For large values of γ, as to be expected, the same forecast times τ_f are produced independent of the chosen initial RAFDA ensemble. The same forecast times τ_f are obtained for ensembles centred around the prior $\boldsymbol{W}_{\mathrm{LR}}$ and for unbiased ensembles for $\log\gamma > 0$. For $\log\gamma < 0$, centring the initial ensemble around the prior provided by LR produces larger forecast times than LR for noisy data, and generally leads to more robust forecast times with respect to the initial spread γ than the unbiased initial ensemble. We remark that the differences between the two initialisation choices depend on the noise level of the training data set; for small noise levels, centring the initial ensemble around the prior provided by LR leads to better forecast times compared to the unbiased initial ensemble for a wider range of values of γ. In the following we will use initial ensembles centred around the prior provided by LR with $\gamma = 1,000$.

Whereas the particular draw of the initial parameters $(\boldsymbol{W}_{\mathrm{in}}, \boldsymbol{b}_{\mathrm{in}})$ can have a large impact on the performance of RAFDA (see also Fig. 7 below), the DA component of RAFDA seems insensitive to the particular random draw of $\boldsymbol{w}_0$. For initial ensembles (34)–(35), which achieve good analysis error of the state variables, we observed that the forecast times do not depend strongly on the particular random draw.

3.4. Distribution of the learned parameters W_{LR} and W_{RAFDA}

Fig. 5 shows how the learned parameters $\boldsymbol{W}_{\mathrm{LR}}$ and $\boldsymbol{W}_{\mathrm{RAFDA}}$ are distributed. Shown are the empirical histograms $\boldsymbol{W}_{\mathrm{LR}}$ and $\boldsymbol{W}_{\mathrm{RAFDA}}$ histogram for a single realisation of a training data set with $N = 40,000$, $D_r = M = 1,000$ and observational noise covariance $\eta = 0.2$. LR yields a forecast time of $\tau_f = 0.9$ and RAFDA a forecast time of $\tau_f = 4.4$. The parameters $\boldsymbol{W}_{\mathrm{LR}}$ are determined by learning all the data at once, with a distribution which is strongly peaked at small values. The parameters $\boldsymbol{W}_{\mathrm{RAFDA}}$ on the other hand are learned sequentially. Surprisingly, we do not see that individual components of $\boldsymbol{W}$ (or its vectorial form $\boldsymbol{w}$) converge, but they converge in distribution. In particular, it is seen that their distribution is near-Gaussian with comparatively much wider support than those obtained by LR. The lack of convergence of the individual entries of $\boldsymbol{W}$ in RAFDA during the assimilation, we believe, is due to the non-uniqueness of the map $\boldsymbol{u}_{n+1} = \boldsymbol{W}\boldsymbol{\phi}(\boldsymbol{u}_n)$, i.e., the same mapping can be achieved (within the error tolerance of the observational noise) for more than one parameter combinations in $\boldsymbol{W}$.
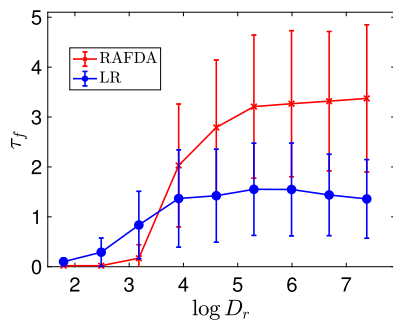
3.5. Dependency on the reservoir dimension D_r

Fig. 6 shows the dependency of the forecast time τ_f on the reservoir dimension D_r for fixed training length $N = 4,000$ and noise level $\eta = 0.2$, averaged over 500 realisations, each involving independently drawn random internal parameters $\boldsymbol{W}_{\mathrm{in}}$ and $\boldsymbol{b}_{\mathrm{in}}$ as well as different training and validation data sets. For each reservoir dimension D_r we choose an ensemble with $M = D_r$ members. The forecast performance of both RAFDA and LR increases with the reservoir dimension D_r initially, with a very steep increase for RAFDA between $\log D_r = 3$ and $\log D_r = 4$, and then saturates for sufficiently large D_r. LR saturates for reservoir dimensions $\log D_r \approx 4$ to $\tau_f \approx 1.4$. RAFDA saturates at larger reservoir dimensions. At saturation RAFDA achieves forecast times $\tau_f \approx 3.4$, more than twice than the corresponding values for classical LR. Note that the mean forecast time τ_f of LR at large reservoir dimensions $D_r \approx 1,500$ ($\log D_r \approx 7.3$) is still smaller with $\tau_f \approx 1.4$ than the mean forecast time of RAFDA for the reservoir dimension $D_r = 50$ ($\log D_r \approx 3.9$) with $\tau_f \approx 2.0$. This suggests that for training it is far more important to control the noisiness of the observations than the expressivity of the model.

Fig. 6 also shows that there is a large spread in the obtained forecast times τ_f caused by outliers in both LR and RAFDA; the spread, relative to the mean forecast time τ_f, however, is diminished by approximately 25% on average for RAFDA compared to LR.
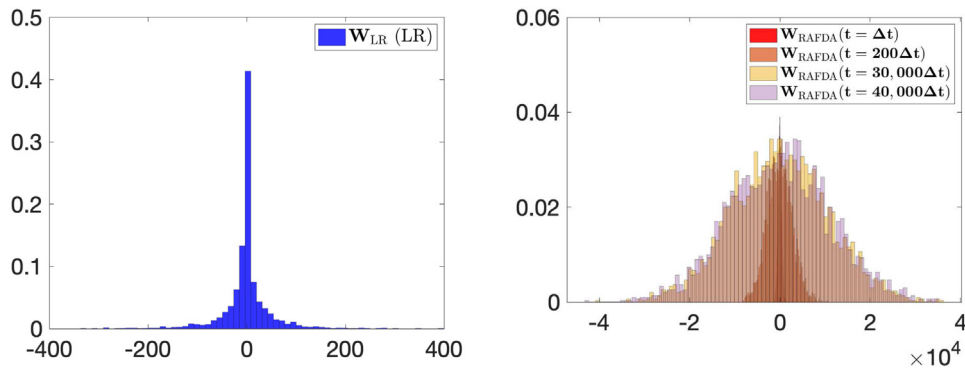
Fig. 5. Empirical histogram of the learned entries of the weight matrix $\boldsymbol{W}_{\mathrm{LR}}$ (left) and $\boldsymbol{W}_{\mathrm{RAFDA}}$ (right). For RAFDA several histograms are shown at different times to illustrate the convergence of the distribution during the assimilation procedure.

Fig. 6. Mean of the forecast time τ_f in units of the Lyapunov time for varying reservoir dimension D_r for RAFDA and for LR, for noise contaminated observations with $\eta = 0.2$. The error bars denote two standard deviations, estimated from 500 independent realisations, differing in their randomly drawn internal parameters ($\boldsymbol{W}_{\mathrm{in}}$, $\boldsymbol{b}_{\mathrm{in}}$), training and validation data sets.

Fig. 7. Empirical histogram of forecast times τ_f in units of the Lyapunov time for $D_r = 1,600$, for noise contaminated observations with $\eta = 0.2$.

To further illustrate the spread of the forecast times, we show in Fig. 7 the empirical histogram of the forecast time τ_f for RAFDA and for LR obtained from 500 independent realisations for a reservoir of size $D_r = 1,600$. RAFDA clearly exhibits improved forecast times with a mean of $\tau_f = 3.4$ over LR with a mean of $\tau_f = 1.4$ Lyapunov time units. Moreover, the large outliers for RAFDA can have forecast times larger than 9 Lyapunov times, whereas extreme cases for LR only reach up to 5.6 Lyapunov times.

In anticipation of the application to higher-dimensional dynamical systems presented in Sections 4 and 5 we state that the reservoir dimension D_r for which the forecast skill saturates depends on the dimension D of the underlying system; the higher the dimension of the dynamical system, the higher the expressivity of the model needs to be.

3.6. Dependency on the training data size N

The size of the training data set needs to be sufficiently large for two distinct reasons. First of all, the training data set needs to sample sufficiently large parts of phase space to allow for an application of the learned mapping to unseen data which may evolve into hitherto unexplored areas of phase space. Secondly, the size of the training data set needs to be sufficiently large to allow for good statistical estimation in both the linear regression as well as the data assimilation components (which has its own initial transient period) and hence to ensure a sufficiently accurate estimation of the mappings (37) and (38), respectively.

Fig. 8 shows the dependency of the mean of the forecast time τ_f on the size N of a noisy training data set with $\eta = 0.2$, averaged

over 500 realisations, again each involving independently drawn random internal parameters $\boldsymbol{W}_{\mathrm{in}}$ and $\boldsymbol{b}_{\mathrm{in}}$ as well as different training and validation data sets. We checked that the analysis fields for the Lorenz-63 system state variables were close to the truth and no ensemble inflation was required. As for the reservoir dimension, the forecast times τ_f increase with increasing length of the training data set N and saturate for sufficiently large training data sets, exhibiting the same twofold improvement of RAFDA over LR as observed in Section 3.5. The slight decline in the forecast time for $\log N > 10$ for LR is consistent with sampling errors, illustrated by the error bars shown in Fig. 8. Both LR and RAFDA suffer from outliers in the prediction which gives rise to the variance in forecast times (see also Fig. 7). We observe that the spread does not decrease with increasing training data length N but saturates in conjunction with the mean forecast time τ_f. As in the dependency on the reservoir dimension, the spread relative to the mean forecast time is reduced on average by 20% for RAFDA compared to LR.

3.7. Dependency on the measurement noise level η

We now test how the forecast capabilities depend on the observational noise level η. Fig. 9 shows the dependency of the forecast time τ_f on the noise level for fixed training data length $N = 4,000$ and reservoir dimension $Dr = 300$, averaged over 500 realisations, each involving independently drawn random internal parameters $\boldsymbol{W}_{\mathrm{in}}$ and $\boldsymbol{b}_{\mathrm{in}}$ as well as different training and validation data sets. For sufficiently small noise levels $\eta < 0.0067$ ($\log \eta < -5$) RAFDA asymptotes to a mean forecast time of $\tau_f \approx 5$ whereas LR asymptotes to only $\tau_f \approx 2$, albeit for a longer range in noise levels. We remark that the regularisation parameter chosen for LR of $\beta = 4 \times 10^{-5}$ was chosen to yield optimal results for
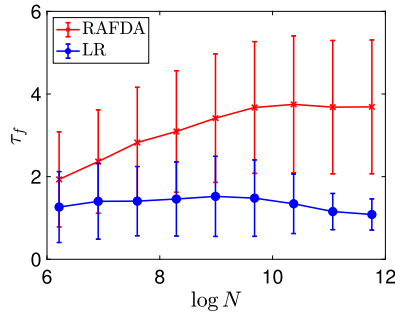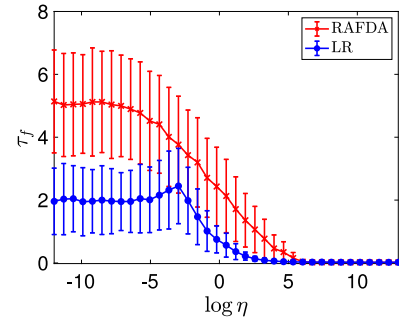
Fig. 8. Mean of the forecast time τ_f in units of the Lyapunov time for varying lengths N of the training data set for RAFDA and for LR, for noise contaminated observations with $\eta = 0.2$. The error bars denote two standard deviations, estimated from 500 independent realisations, differing in their randomly drawn internal parameters ($\boldsymbol{W}_{\mathrm{in}}, \boldsymbol{b}_{\mathrm{in}}$), training and validation data sets.

Fig. 9. Mean of the forecast time τ_f in units of the Lyapunov time for varying noise levels η for RAFDA and for LR. The error bars denote two standard deviations, estimated from 500 independent realisations, differing in their randomly drawn internal parameters ($\boldsymbol{W}_{\mathrm{in}}, \boldsymbol{b}_{\mathrm{in}}$), training and validation data sets.

fixed $\eta = 0.2$. However, for $\beta \to 0$ which is optimal for LR for $\eta = 0$ the maximal forecast time is $\tau_f \approx 3.8$ (cf Fig. 3), below the corresponding value of RAFDA with $\tau_f \approx 5$. The range of constant forecast times is followed upon increasing the noise level by an exponentially decaying range, before the models lose any forecast skill with $\tau_f \to 0$. The exponential behaviour of the forecast time τ_f with respect to the observational noise strength η is consistent with the sensitivity of chaotic dynamical systems with respect to their initial condition and the exponential separation of nearby trajectories. To corroborate this interpretation we have confirmed that the same forecast times are achieved when simultaneously doubling the data size to $N = 8,000$ and the noise level η.

The forecast time τ_f varies non-monotonically with the observational noise strength η for LR. We observe an increase in τ_f for $\log \eta \approx -3$. The value of the observational noise η for which this increase in τ_f occurs depends on β and decreases for decreasing values of β (not shown). Recall that LR has been devised under the assumption of model error, with covariance proportional to β, without consideration of eventual observational error. It seems that a time series with a specified observational noise can be approximated by an equivalent model error, and hence allows for an adequate application of LR for this particular observational noise strength.

To further illustrate that standard random feature maps and LR are not suited to learn the dynamics from noisy observations, we tested that simple denoising does not lead to an improved forecasting skill of LR. By applying a simple denoising algorithm of a moving average whereby each observation $\boldsymbol{u}^{\mathrm{o}}(t_i)$ is replaced by $\frac{1}{3}(\boldsymbol{u}^{\mathrm{o}}(t_i - \Delta t) + \boldsymbol{u}^{\mathrm{o}}(t_i) + \boldsymbol{u}^{\mathrm{o}}(t_i + \Delta t))$ to average out the noisy signal [56], yields a mean forecast time of $\tau_f = 1.6$ for $\eta = 0.2$, compared to $\tau_f = 3.3$ for RAFDA, when averaged over 500 realisations as above. Similarly, the RMS error at lead time $t = 1$ is 9.1 compared to 3.9 for RAFDA. It is likely that a more refined denoising scheme could boost the performance of LR; but such a scheme would be problem dependent and would require additional tuning.

The reader may wonder why RAFDA outperforms LR for $\eta \to 0$. If the output data were lying in the span of the random feature maps, then linear regression would be the optimal solution minimising (13) with $\mathcal{L}(\boldsymbol{W}_{\mathrm{LR}}) = 0$. We remark that LR with $\beta > 0$ assumes an intrinsic model error (16) which is absent from our deterministic Lorenz-63 system. We have tested for $\beta = 10^{-11}$ and several larger values of β that $0 < \mathcal{L}(\boldsymbol{W}_{\mathrm{LR}}) < \mathcal{L}(\boldsymbol{W}_{\mathrm{RAFDA}})$ for the training data but nevertheless RAFDA generates surrogate models which consistently exhibit larger forecast times τ_f when applied to unseen data (not shown). Hence, despite the fact that RAFDA performs suboptimally in minimising the cost function

(13) it generalises much better to unseen data. The fact that the EnKF does not yield optimal estimates for nonlinear problems has long been known (see, for example, [57]).

3.8. Application to ensemble forecasting

In chaotic dynamical systems a single forecast is to a certain degree meaningless as small uncertainties in the initial condition can lead to widely differing forecasts at a later time. Probabilistic forecasts provide a more appropriate framework for forecasting. In particular, ensemble forecasting, whereby a Monte Carlo estimate of the underlying probability density function is provided by running an ensemble of initial conditions, are now widely used in numerical weather forecasting issuing both the most probable forecast and its associated uncertainty [58–60]. The quality of such ensemble forecasts crucially depends on how the ensembles are generated. There exist several strategies using singular vectors [61,62], bred vectors [63–65], analysis ensembles from ensemble Kalman filters [66,67], and more recently analogs [68]. Here we show that ensembles obtained from RAFDA provide reliable forecast ensembles to be used in probabilistic forecasts. A good probabilistic forecasts is not necessarily one with the smallest root-mean-square (RMS) error (consider a probability density function with disjoint support, then the mean may not be even an actual physical state), but rather one where each ensemble member has equal probability of being closest to the truth. Such ensembles are called reliable. We probe the reliability here with the continuous ranked probability score (CRPS) [69,70]. The CRPS is defined for a lead time τ as

$$\mathrm{CRPS}(\tau) = \frac{1}{D} \sum_{k=1}^{D} \int_{-\infty}^{\infty} \Big[F(u; \tau, k) - F_{\mathrm{truth}}(u; \tau, k) \Big]^2 \, du, \qquad (39)$$

where F and F_{truth} are the cumulative probability distributions of the forecast ensemble and truth, respectively, which are estimated as

$$F(u; \tau, k) = \frac{1}{M} \sum_{m=1}^{M} \Theta \Big(u - u^{(m,k)}(\tau) \Big), \qquad (40)$$

and

$$F_{\mathrm{truth}}(u; \tau, k) = \Theta \Big(u - u_{\mathrm{truth}}^{(k)}(\tau) \Big), \qquad (41)$$

where $\Theta(x)$ is the Heaviside function with $\Theta(x) = 0$ for $x < 0$ and $\Theta(x) = 1$ otherwise, and $u^{(m,k)}(\tau)$ denotes the k-component of the mth ensemble member $\boldsymbol{u}^{(m)}(\tau)$ at forecast time $\tau > 0$. Smaller values of CRPS indicate better reliability.

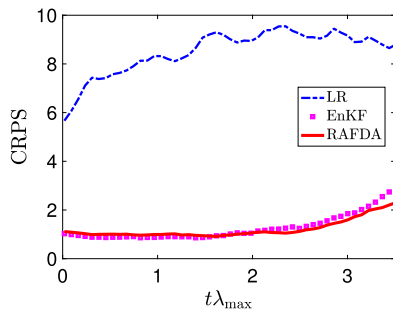We consider here ensembles obtained in an EnKF data assimilation where the forecast model is given by the surrogate

Fig. 10. CRPS as a function of the lead time τ for RAFDA, LR and EnKF ensembles.

model obtained from either RAFDA or from LR. Both surrogate models and their associated weight matrices were obtained prior to and independently from the data assimilation cycles using a training set of length $N = 250,000$, consisting of noisy observations with $\eta = 0.2$ sampled every $\Delta t = 0.02$ model time units. The LR and RAFDA models are kept fixed and no further parameter estimation occurs during the data assimilation. We use an observation time of $\Delta t = 0.02$ for the independent data assimilation cycles used to generate the ensembles; the same time interval used for training both LR and RAFDA. We further compare with results obtained from a traditional EnKF ensemble where the full Lorenz-63 model (36) is employed as forecast model. We perform 500 independent analysis cycles, allowing for each a burn-in period of 2,000 analysis cycles before the analysis error and the CRPS are evaluated. RAFDA performs equally well as the full EnKF which uses the full Lorenz-63 model (36) as forecast model, with analysis errors of 2.4 and 2.3, respectively. The LR model was not able to produce good analysis with a remarkably larger analysis error of 11.0. Fig. 10 shows CRPS(τ), averaged over the 500 independent realisations. Whereas LR ensembles cannot be classified as reliable, RAFDA ensembles are shown to be as reliable as EnKF ensembles.

4. Partial differential equations: Kuramoto–Sivashinsky equation

We now consider artificially generated time series obtained from a numerical simulation of the Kuramoto–Sivashinsky equation [71–74]

$$u_t + uu_x + u_{xx} + u_{xxxx} = 0 \qquad (42)$$

with periodic boundary conditions. For system length $L = 22$ the Kuramoto–Sivashinsky equation is chaotic with a maximal Lyapunov exponent of $\lambda_{max} = 0.043$ [75]. We remark that although the Kuramoto–Sivashinsky equation is a partial-differential equation, its dynamics evolves on a finite dimensional manifold [76].

We generate observations $\boldsymbol{u}_n^o$ by integrating (42) using a pseudo-spectral Crank–Nicolson scheme where the nonlinearity is treated with a second-order Adams–Bashforth scheme. We employ a temporal discretisation step of $\delta t = 0.001$ and use 64 spatial grid points, and sample every $\Delta t = 0.25$ time units to obtain observations $\boldsymbol{u}_n^o \in \mathbb{R}^{64}$. An initial transient period of 25×10^6 time units is discarded to ensure that the dynamics has settled onto the attractor.

The internal weight matrix and bias are chosen randomly according to (30) with $w = 0.1$ and $b = 1$ with a reservoir of size $D_r = 2,000$. We consider here only a single realisation of the internal parameters ($\boldsymbol{W}_{in}, \boldsymbol{b}_{in}$). We train RAFDA on a noisy training data set of length $N = 70,000$ with $\eta = 0.01$. The data assimilation component of RAFDA was executed with $M = 1,000$ ensemble members and without any inflation with an

initial ensemble chosen according to (35) with $\gamma = 6.4 \cdot 10^{-6}$. Fig. 11 depicts results of the RAFDA forecast showing that reasonable forecasts can be made up to four Lyapunov times. Classical random feature maps with linear ridge regression trained on the same data set only yield forecast horizons of one Lyapunov time (not shown). If LR is trained on noiseless data, it achieves comparable forecast skill to RAFDA (not shown), again illustrating that LR has difficulties with training on noisy data.

To explore the capabilities of RAFDA to deal with higher-dimensional systems more careful numerical simulations would be needed. We have refrained here from optimising the reservoir weights w and b and the reservoir dimension D_r to achieve larger forecast horizons, and have only shown a single realisation which shows that higher-dimensional problems can be addressed within our framework.

5. Closure models: Multi-scale Lorenz-96 system

We consider now the problem of model closure in multi-scale systems. In multi-scale systems one is typically only interested in the dynamics of the large-scale slow variables. Moreover, one typically only has access to observations of the resolved slow dynamics. The objective of model closure, also known as the subgrid-scale parametrisation problem, is to find a self-consistent dynamical model for the resolved slow large-scale variables. The effective model for the slow large-scale dynamics allows for a larger time step in the numerical simulation compared to the prohibitively small time steps required in the full stiff multi-scale system.

We consider here the multi-scale Lorenz-96 system for K slow variables $X^{(k)}$ which are each coupled to J fast variables $Y^{(j,k)}$, given by

$$\frac{d}{dt}X^{(k)} = -X^{(k-1)}(X^{(k-2)} - X^{(k+1)}) - X^{(k)} + F - \frac{hc}{d}\sum_{j=1}^{J}Y^{(j,k)},$$
$$(43)$$

$$\frac{d}{dt}Y^{(j,k)} = -cbY^{(j+1,k)}(Y^{(j+2,k)} - Y^{(j-1,k)}) - cY^{(j,k)} + \frac{hc}{d}X^{(k)}, \quad (44)$$

with cyclic boundary conditions $X^{(k+K)} = X^{(k)}$, $Y^{(j,k+K)} = Y^{(j,k)}$ and $Y^{(j+J,k)} = Y^{(j,k+1)}$, giving a total of $D = K(J + 1)$ variables. This set of equations was introduced as a caricature for the midlatitude atmosphere dynamics [77]. The variables $X^{(k)}$ model large scale atmospheric fields arranged on a latitudinal circle, such as synoptic weather systems. Each of the $X^{(k)}$ variables is connected to J small-scale variables $Y^{(j,k)}$. The time scale separation is parametrised by the coefficient c, and the ratio of the amplitudes of the large-scale and the small-scale variables is controlled by b. The coupling between the slow and fast variables is controlled by the parameter h. Both large-scale and small-scale variables evolve, when uncoupled, according to nonlinear transport and linear damping; the large-scales are additionally subjected to external forcing F. We choose here as parameters $K = 8$ and $J = 32$, leading to a total of 256 variables, and $F = 20$, $c = d = 10$ and $h = 1$ as in [78,79]. The choice $c = d = 10$ implies that the fast variables $Y^{(j,k)}$ fluctuate with a 10 times higher frequency and with an approximately 10 times smaller amplitude when compared to the slow variables $X^{(k)}$. Setting the coupling constant $h = 1$, corresponds to strong coupling where the dynamics is driven by the fast sub-system [80].

Our aim is to provide a reduced model of the form

$$\frac{d}{dt}X^{(k)} = G^{(k)}(\boldsymbol{X}) + \psi(X^{(k)}), \qquad (45)$$

with $\boldsymbol{X} = (X^{(1)}, \ldots, X^{(K)})^{\mathsf{T}}$, $G^{(k)}(\boldsymbol{X}) = -X^{(k-1)}(X^{(k-2)} - X^{(k+1)}) - X^{(k)} + F$, and where the closure term $\psi(x)$ parametrises the effect
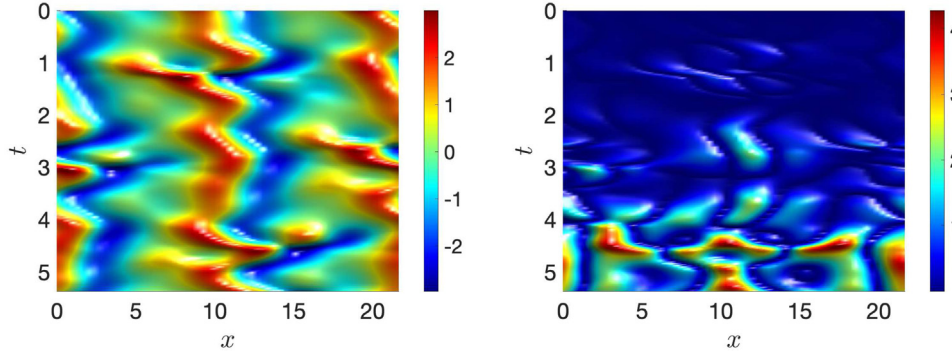
Fig. 11. Hovmöller diagram of the forecast field $u(x,t)$ of the Kuramoto–Sivashinsky equation (42). Left: RAFDA forecast. Right: Root-mean-square error of the RAFDA forecast with respect to the truth. Times are in units of $1/\lambda_{\max}$.

of the fast unresolved dynamics. Here the closure term $\psi(x)$ will be learned by RAFDA. Note that one and the same closure map can be applied at all grid points due to the spatial homogeneity of the problem. The application we have in mind is that a scientist has available a physics-based model for the resolved slow large-scale dynamics, i.e. the general form of Eq. (45) and the expression for the resolved vectorfield $G^{(k)}(\boldsymbol{X})$, but does not have access to the unresolved fast small-scale model, and the closure model needs to be inferred from noisy observations.

In the case when the full multi-scale model (43)–(44) is available and noiseless data of both the slow and fast variables are available, the closure problem has been attempted outside the scope of machine learning in [78,79]. Therein deterministic parametrisations were proposed by data-fitting of the closure term to the resolved variables with polynomials to obtain

$$\psi_{\text{Wilks}}(x) = 0.262 + 1.45x - 0.0121x^2 - 0.00713x^3 + 0.000296x^4 \tag{46}$$

$$\psi_{\text{AMP}}(x) = 0.341 + 1.3x - 0.0136x^2 - 0.00235x^3. \tag{47}$$

Here we assume we are only given noisy observations of the slow variables $\boldsymbol{X}_n$ at discrete times $t_n = n\Delta t$. In LR, we set $\boldsymbol{u}_n = \boldsymbol{X}_n$, $n = 0, \ldots, N$, and the closure mapping $\psi(x)$ is learned directly from the data via finite-differencing. More specifically, the closure is performed by minimising the difference between the random feature map approximation of the closure term $\psi(x) = \phi(x)$ over all grid points $x = X^{(k)}$, and the approximation of the closure term provided by the observations

$$\frac{X_n^{(k)} - X_{n-1}^{(k)}}{\Delta t} - G^{(k)}(\boldsymbol{X}_{n-1}).$$

This finite differencing, however, constitutes a bad estimator of the closure term $\psi(x)$ for noisy observations of the slow variables. In the following we therefore only present results for LR in the case of zero measurement error (i.e. the same setting in which the polynomial fits (46) and (47) were obtained). RAFDA, on the other hand, employs as the forecast model within the data assimilation training phase the reduced model (45) where $\psi(x)$ is estimated using random feature maps $\psi(x) = \boldsymbol{W} \tanh(\boldsymbol{W}_{\text{in}}x + \boldsymbol{b}_{\text{in}})$ (cf. (10) and (11)) where $\boldsymbol{W} \in \mathbb{R}^{1 \times D_r}$, $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{D_r \times 1}$, and $\boldsymbol{b}_{\text{in}} \in \mathbb{R}^{D_r}$. RAFDA is trained on $N = 100,000$ noisy observations of all components $X^{(k)}$ of $\boldsymbol{X}$ with variance $\eta = 0.02$ sampled at $\Delta t = 0.0005$, and an initial ensemble chosen according to (35) with $\gamma = 0.01$, centred around the prior provided by LR. We employ inflation $\alpha = 1 + 0.1\Delta t$. We chose a reservoir of size $D_r = 1,000$ and draw the internal parameters using $w = 0.05$ and $b = 1$. The validation is performed by propagating the reduced model (45) with the learned mapping $\psi(x)$ added at each grid point in an
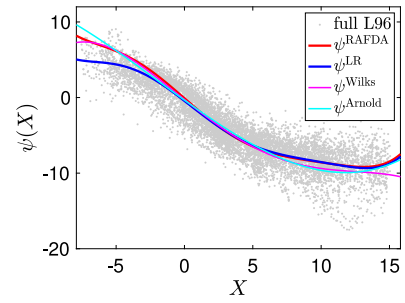
Fig. 12. Closure terms $\psi(x)$ for the multi-scale L96-system (43)–(44) estimated using various parametrisations.

Table 1
Relative error of the mean μ and the variance σ^2 of the slow variables $\boldsymbol{X}$ for the various parametrisation schemes.

	μ	σ^2
RAFDA	0.0056	0.019
LR	0.029	0.028
Wilks	0.033	0.065
Arnold et al.	0.016	0.063

Euler time-stepping method with step-size $\delta t = 0.005$ for a total of 10^6 time steps.

Fig. 12 shows the closure term $\psi(x)$ estimated from RAFDA, which was trained with noisy data with $\eta = 0.02$, and from LR and from the polynomial fits (46)–(47), which were all obtained from noise free data. Table 1 lists the associated relative errors of the mean and the variance of $\boldsymbol{X}$ for the various parametrisation schemes used. The closure model provided by RAFDA outperforms both the closure of LR and the ones given by (46)–(47). We show in Fig. 13 the empirical probability density function of a validation time series, created independently of the training data set. It is seen that the reference empirical probability density function of the full multi-scale L96 system assigns more probability to higher values of $\boldsymbol{X}$. We conjecture that this underdispersiveness of the closure models stems from their assumed deterministic nature. The deterministic closures schemes considered here ignore the diffusive effect of the fast variables which arises for finite time-scale and spatial-scale separation as fluctuations around the deterministic mean behaviour. As a first step towards a full stochastic parametrisation one could use RAFDA to learn the propagator of the mean $E^{(k)} = \sum_{j=1}^{J} Y^{(j,k)}$ alongside the mapping $\psi(x)$.
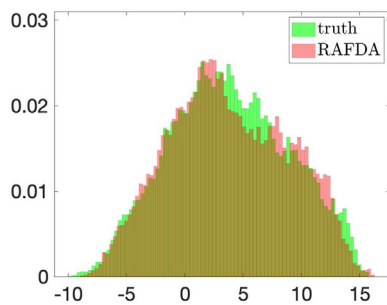
Fig. 13. Empirical probability density function of the slow variables X_k of the full multi-scale L96-system (43)–(44) and of the RAFDA closure model.

6. Discussion and outlook

We have proposed a new data-driven physics-agnostic forecast model, coined RAFDA, which combines random feature maps with DA. The method is designed to provide cheap surrogate mappings to be used in forecasting or to construct closure models when no *a priori* knowledge of the evolution equations is available and the underlying system is only accessible via noisy observations. The linear coefficients of the random feature map are sequentially updated incorporating incoming observations using an EnKF, which itself uses the random feature map as its forecast model. We have tested the forecast capabilities in several situations for ordinary and partial differential equations. For the Lorenz-63 model and the Kuramoto–Sivashinsky equation we were able to achieve a forecast horizon of several Lyapunov times. We investigated the influence of the data length, the number of feature maps and the noise strength on the forecast horizon. Depending on the dimension of the underlying dynamical system a larger number of random feature maps and larger training data sets are required. Once trained, however, the computational cost to run the surrogate model can be significantly cheaper than numerical integration of the original full model. For example, for the Kuramoto–Sivashinsky model we used a discretisation time step of $\delta t = 10^{-3}$ whereas the surrogate RAFDA model was trained on a time series with $\Delta t = 0.25$, which implies a computational gain in running time of $\Delta t / \delta t = 250$. However, we stress that the real advantage of RAFDA becomes apparent when the model is unknown.

Besides providing remarkable forecast skills for individual trajectories, we showed that ensembles generated in data assimilation cycles which use a pre-trained RAFDA forecast model are reliable which makes them attractive candidates for probabilistic forecasting where one would like to issue an uncertainty quantification of a forecast. Furthermore, we showed that in the multi-scale Lorenz-96 model RAFDA allowed us to find a Markovian closure model for the resolved variables having only observed its slow variables.

We remark that our methodology of sequentially updating the parameters with the incoming observations using data assimilation is not restricted to EnKFs, but one may instead employ other linear or nonlinear filtering methods such as [81]; equally, to estimate the parameters W one is not restricted to linear regression but may employ nonlinear regression methods if desired.

An open question is how to choose the random parameters of the feature map. The universal approximation property ensures that any continuous function can be well approximated by a linear combination of the features, but it does not provide any statement on how to choose the hyper-parameters and on how many features are needed to achieve good approximation. We have shown that the forecasts capabilities sensitively depend on the random hyper-parameters and that parameters which lead to good approximation in the noiseless case may fail to approximate the forecast map when trained on noisy data. A natural extension is to determine the optimal parameters W_{in} and b_{in} simultaneously with W, possibly using again an ensemble Kalman filter. Rather than random features which are related to RKHS, the machine-learning framework would then consist of a two-layer network [45–49]. This extension is planned for further research.

Further, recent developments on operator-valued kernels in the context of random feature maps has shown that the number of learnable parameters can be reduced to D_r, independent of the dimension D of the underlying system [39,82]. The methodology developed in this paper easily generalises to such vector- or operator-valued random feature maps. However, it remains to be investigated whether or not the required number D_r of feature maps remains indeed independent of the dimension D of the underlying dynamical systems unless the random feature maps are specifically designed for the problem at hand. This poses an interesting question for further research, in particular in view of dealing with high-dimensional systems. The same applies to the application of localisation to the EnKF. Provided more information is available on the correlation structure of the unknown parameters W, a more refined localisation procedure could be employed.

The numerical experiments in this paper have all been conducted under an ideal twin setting where the underlying dynamics is deterministic and no random or systematic model errors had to be considered. The robustness of the proposed RAFDA with regard to such random or systematic perturbations will be the subject of future work. The same applies to a more systematic and theoretical investigation of the impact of the various algorithmic parameters on the behaviour of RAFDA within the ideal twin setting and beyond.

CRediT authorship contribution statement

Georg A. Gottwald: Methodology, Simulations, Writing. **Sebastian Reich:** Methodology, Writing.

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

References

[1] J. Han, A. Jentzen, W. E, Solving high-dimensional partial differential equations using deep learning, Proc. Natl. Acad. Sci. 115 (34) (2018) 8505–8510, http://dx.doi.org/10.1073/pnas.1718942115.

[2] J. Han, C. Ma, Z. Ma, W. E, Uniformly accurate machine learning-based hydrodynamic models for kinetic equations, Proc. Natl. Acad. Sci. 116 (44) (2019) 21983–21991, http://dx.doi.org/10.1073/pnas.1909854116.

[3] M. Raissi, G.E. Karniadakis, Hidden physics models: Machine learning of nonlinear partial differential equations, J. Comput. Phys. 357 (2018) 125–141, http://dx.doi.org/10.1016/j.jcp.2017.11.039.

[4] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707, http://dx.doi.org/10.1016/j.jcp.2018.10.045.

[5] A. Rahimi, B. Recht, Random features for large-scale kernel machines, in: J.C. Platt, D. Koller, Y. Singer, S.T. Roweis (Eds.), Advances in Neural Information Processing Systems 20, Curran Associates, Inc., 2008, pp. 1177–1184, URL http://papers.nips.cc/paper/3182-random-features-for-large-scale-kernel-machines.pdf.

[6] J. Park, I. Sandberg, Universal approximation using radial-basis-function networks, Neural Comput. 3 (1991) 246–257, http://dx.doi.org/10.1162/neco.1991.3.2.246.

[7] G. Cybenko, Approximation by superposition of a sigmoidal function, Math. Contr. Sign. Syst. 2 (1989) 303–314, http://dx.doi.org/10.1007/BF02551274.

[8] A. Barron, Universal approximation bounds for superposition of a sigmoidal function, IEEE Trans. Inform. Theory 39 (1993) 930–945, http://dx.doi.org/10.1109/18.256500.

[9] A. Rahimi, B. Recht, Uniform approximation of functions with random bases, in: 2008 46th Annual Allerton Conference on Communication, Control, and Computing, 2008, pp. 555–561.

[10] F. Bach, Breaking the curse of dimensionality with convex neural networks, J. Mach. Learn. Res. 18 (19) (2017) 1–53, URL http://jmlr.org/papers/v18/14-546.html.

[11] F. Bach, On the equivalence between kernel quadrature rules and random feature expansions, J. Mach. Learn. Res. 18 (21) (2017) 1–38, URL http://www.jmlr.org/papers/v18/15-178.html.

[12] Y. Sun, A. Gilbert, A. Tewari, On the approximation capabilities of ReLU neural networks and random ReLU features, 2019, arXiv:1801.04374.

[13] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Comput. 14 (11) (2002) 2531–2560, http://dx.doi.org/10.1162/089976602760407955.

[14] H. Jaeger, A Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and the "Echo State Network" Approach, GMD-Report 159, German National Research Institute for Computer Science, 2002.

[15] H. Jaeger, H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science 304 (5667) (2004) 78–80, http://dx.doi.org/10.1126/science.1091277.

[16] J. Pathak, B. Hunt, M. Girvan, Z. Lu, E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach, Phys. Rev. Lett. 120 (2018) 024102, http://dx.doi.org/10.1103/PhysRevLett.120.024102.

[17] A. Majda, J. Harlim, Filtering Complex Turbulent Systems, Cambridge University Press, Cambridge, 2012.

[18] K. Law, A. Stuart, K. Zygalakis, Data Assimilation: A Mathematical Introduction, Springer-Verlag, New York, 2015.

[19] S. Reich, C. Cotter, Probabilistic Forecasting and Bayesian Data Assimilation, Cambridge University Press, New York, 2015, p. x+297.

[20] F. Hamilton, T. Berry, T. Sauer, Ensemble Kalman filtering without a model, Phys. Rev. X 6 (2016) 011021, http://dx.doi.org/10.1103/PhysRevX.6.011021.

[21] H. Abarbanel, P. Rozdeba, S. Shirman, Machine learning: Deepest learning as statistical data assimilation problems, Neural Comput. 30 (2018) 2025–2055, http://dx.doi.org/10.1162/neco_a_01094.

[22] D. Nguyen, S. Ouala, L. Drumetz, R. Fablet, EM-like learning chaotic dynamics from noisy and partial observations, 2019, arXiv:1903.10335.

[23] M. Bocquet, J. Brajard, A. Carrassi, L. Bertino, Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models, Nonlinear Process. Geophys. 26 (2019) 143–162, http://dx.doi.org/10.5194/npg-26-143-2019.

[24] M. Bocquet, J. Brajard, A. Carrassi, L. Bertino, Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization, Found. Data Sci. 2 (2020) 55–80, http://dx.doi.org/10.3934/fods.2020004.

[25] J. Brajard, A. Carrassi, M. Bocquet, L. Bertino, Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the Lorenz 96 model, J. Comput. Sci. 44 (2020) 101171, http://dx.doi.org/10.1016/j.jocs.2020.101171.

[26] M. Bocquet, A. Farchi, Q. Malartic, Online learning of both state and dynamics using ensemble Kalman filters, Found. Data Sci. (2020) http://dx.doi.org/10.3934/fods.2020015.

[27] N. Kantas, A. Doucet, S.S. Singh, J. Maciejowski, N. Chopin, On particle methods for parameter estimation in state-space models, Stat. Sci. 30 (2015) 328–351, http://dx.doi.org/10.1214/14-STS511.

[28] P.D. Dueben, P. Bauer, Challenges and design choices for global weather and climate models based on machine learning, Geosci. Model Dev. 11 (10) (2018) 3999–4009, http://dx.doi.org/10.5194/gmd-11-3999-2018, URL https://gmd.copernicus.org/articles/11/3999/2018/.

[29] S. Rasp, M.S. Pritchard, P. Gentine, Deep learning to represent subgrid processes in climate models, Proc. Natl. Acad. Sci. 115 (39) (2018) 9684–9689, http://dx.doi.org/10.1073/pnas.1810286115, URL https://www.pnas.org/content/115/39/9684.

[30] D.J. Gagne II, H.M. Christensen, A.C. Subramanian, A.H. Monahan, Machine learning for stochastic parameterization: Generative adversarial networks in the Lorenz '96 model, J. Adv. Modelling Earth Syst. 12 (3) (2020) http://dx.doi.org/10.1029/2019MS001896, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019MS001896.

[31] T. Bolton, L. Zanna, Applications of deep learning to ocean data inference and subgrid parameterization, J. Adv. Modelling Earth Syst. 11 (1) (2019) 376–399, http://dx.doi.org/10.1029/2018MS001472, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018MS001472.

[32] J. Brajard, A. Carrassi, M. Bocquet, L. Bertino, Combining data assimilation and machine learning to infer unresolved scale parametrisation, 2020, arXiv:2009.04318.

[33] B. Nadiga, Reservoir computing as a tool for climate predictability studies, J. Adv. Modelling Earth Syst. n/a (n/a) (2020) http://dx.doi.org/10.1029/2020MS002290, e2020MS002290, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020MS002290.

[34] D. Qi, A.J. Majda, Using machine learning to predict extreme events in complex systems, Proc. Natl. Acad. Sci. 117 (2020) 52–59, http://dx.doi.org/10.1073/pnas.1917285117.

[35] G. Evensen, Data Assimilation: The Ensemble Kalman Filter, Springer, New York, 2006.

[36] C.M. Bishop, Pattern Recognition and Machine Learning, in: Information Science and Statistics, Springer, New York, 2006.

[37] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, in: Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA, 2006, p. xviii+248.

[38] W. E., C. Ma, S. Wojtowytsch, L. Wu, Towards a Mathematical Understanding of Neural Network-Based Machine Learning: What we know and what we don't, 2020, arXiv:2009.10713.

[39] N. Nelsen, A. Stuart, The random feature model for input-ouput maps between Banach spaces, 2020, arXiv:2005.10224.

[40] M.O. Williams, I.G. Kevrekidis, C.W. Rowley, A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition, J. Nonlinear Sci. 25 (2015) 1307–1346, http://dx.doi.org/10.1007/s00332-015-9258-5.

[41] M.S. Hemati, C.W. Rowley, E.A. Deem, L.N. Cattafesta, De-biasing the dynamic mode decomposition for applied Koopman spectral analysis of noisy datasets, Theor. Comput. Fluid Dyn. 31 (2017) 349–368, http://dx.doi.org/10.1007/s00162-017-0432-2.

[42] 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. 113 (15) (2016) 3932–3937, http://dx.doi.org/10.1073/pnas.1517384113.

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

[44] G. Burgers, P.J. van Leeuwen, G. Evensen, Analysis scheme in the ensemble Kalman filter, Mon. Weather Rev. 126 (6) (1998) 1719–1724, http://dx.doi.org/10.1175/1520-0493(1998)126<1719:ASITEK>2.0.CO;2.

[45] L. Chizat, F. Bach, On the global convergence of gradient descent for over-parameterized models using optimal transport, Adv. Neural Inf. Process. Syst. (2018) 3036–3046, URL https://proceedings.neurips.cc/paper/2018/file/a1afc58c6ca9540d057299ec3016d726-Paper.pdf.

[46] S. Mei, A. Montanari, P.-M. Nguyen, A mean field view of the landscape of two-layer neural networks, Proc. Natl. Acad. Sci. USA 115 (33) (2018) E7665–E7671, http://dx.doi.org/10.1073/pnas.1806579115.

[47] G.M. Rotskoff, E. Vanden-Eijnden, Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error, 2018, arXiv:1805.00915.

[48] J. Sirignano, K. Spiliopoulos, Mean field analysis of neural networks: A law of large numbers, SIAM J. Appl. Math. 80 (2) (2020) 725–752, http://dx.doi.org/10.1137/18M1192184.

[49] W. E, C. Ma, L. Wu, A priori estimates of the population risk for two-layer neural networks, Commun. Math. Sci. 17 (5) (2019) 1407–1425, http://dx.doi.org/10.4310/CMS.2019.v17.n5.a11.

[50] P.L. Houtekamer, F. Zhang, Review of the ensemble Kalman filter for atmospheric data assimilation, Mon. Weather Rev. 144 (12) (2016) 4489–4532, http://dx.doi.org/10.1175/MWR-D-15-0440.1.

[51] P.L. Houtekamer, H.L. Mitchell, Data assimilation using an ensemble Kalman filter technique, in: Monthly Weather Review, Mon. Weather Rev. 126 (3) (1998) 796–811, http://dx.doi.org/10.1175/1520-0493(1998)126<0796:DAUAEK>2.0.CO;2.

[52] P.L. Houtekamer, H.L. Mitchell, A sequential ensemble Kalman filter for atmospheric data assimilation, Mon. Weather Rev. 129 (1) (2001) 123–136, http://dx.doi.org/10.1175/1520-0493(2001)129<0123:ASEKFF>2.0.CO;2.

[53] T.M. Hamill, J.S. Whitaker, C. Snyder, Distance-dependent filtering of background covariance estimates in an ensemble Kalman filter, Mon. Weather Rev. 129 (11) (2001) 2776–2790, http://dx.doi.org/10.1175/1520-0493(2001)129<2776:DDFOBE>2.0.CO;2.

[54] J.L. Anderson, S.L. Anderson, A Monte Carlo implementation of the non-linear filtering problem to produce ensemble assimilations and forecasts, Mon. Weather Rev. 127 (12) (1999) 2741–2758, http://dx.doi.org/10.1175/1520-0493(1999)127<2741:AMCIOT>2.0.CO;2.

[55] E.N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. 20 (2) (1963) 130–141, http://dx.doi.org/10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

[56] H. Kantz, T. Schreiber, Nonlinear Time Series Analysis, second ed., Cambridge University Press, Cambridge, 2004, p. xvi+369.

[57] F. Le Gland, V. Monbet, V.-D. Tran, Large sample asymptotics for the ensemble Kalman filter, in: D. Crisan, B. Rozovskii (Eds.), Handbook on Nonlinear Filtering, Oxford University Press, 2011, pp. 598–631.

[58] E.S. Epstein, Stochastic dynamic prediction, Tellus 21 (6) (1969) 739–759, http://dx.doi.org/10.1111/j.2153-3490.1969.tb00483.x.

[59] C.E. Leith, Theoretical skill of Monte Carlo forecasts, Mon. Weather Rev. 102 (6) (1974) 409–418, http://dx.doi.org/10.1175/1520-0493(1974)102<0409:TSOMCF>2.0.CO;2.

[60] M. Leutbecher, T. Palmer, Ensemble forecasting, J. Comput. Phys. 227 (7) (2008) 3515–3539, http://dx.doi.org/10.1016/j.jcp.2007.02.014.

[61] E.N. Lorenz, A study of the predictability of a 28-variable atmospheric model, Tellus 17 (3) (1965) 321–333, http://dx.doi.org/10.3402/tellusa.v17i3.9076.

[62] T.N. Palmer, Extended-range atmospheric prediction and the Lorenz model, Bull. Am. Meteorol. Soc. 74 (1) (1993) 49–66, http://dx.doi.org/10.1175/1520-0477(1993)074<0049:ERAPAT>2.0.CO;2.

[63] Z. Toth, E. Kalnay, Ensemble forecasting at NMC: The generation of perturbations, Bull. Am. Meteorol. Soc. 74 (12) (1993) 2317–2330, http://dx.doi.org/10.1175/1520-0477(1993)074<2317:EFANTG>2.0.CO;2.

[64] Z. Toth, E. Kalnay, Ensemble forecasting at NCEP and the breeding method, Mon. Weather Rev. 125 (12) (1997) 3297–3319, http://dx.doi.org/10.1175/1520-0493(1997)125<3297:EFANAT>2.0.CO;2.

[65] B. Giggins, G.A. Gottwald, Stochastically perturbed bred vectors in multi-scale systems, Q. J. R. Meteorol. Soc. 145 (2019) 642–658, http://dx.doi.org/10.1002/qj.3457.

[66] X. Wang, C.H. Bishop, A comparison of breeding and ensemble trans-form Kalman filter ensemble forecast schemes, J. Atmos. Sci. 60 (9) (2003) 1140–1158, http://dx.doi.org/10.1175/1520-0469(2003)060<1140:ACOBAE>2.0.CO;2.

[67] R. Buizza, P.L. Houtekamer, G. Pellerin, Z. Toth, Y. Zhu, M. Wei, A comparison of the ECMWF, MSC, and NCEP global ensemble prediction systems, Mon. Weather Rev. 133 (5) (2005) 1076–1097, http://dx.doi.org/10.1175/MWR2905.1.

[68] A. Atencia, I. Zawadzki, Analogs on the lorenz attractor and ensemble spread, Mon. Weather Rev. 145 (4) (2017) 1381–1400, http://dx.doi.org/10.1175/MWR-D-16-0123.1.

[69] H. Hersbach, Decomposition of the continuous ranked probability score for ensemble prediction systems, Weather Forecast. 15 (5) (2000) 559–570, http://dx.doi.org/10.1175/1520-0434(2000)015<0559:DOTCRP>2.0.CO;2.

[70] D.S. Wilks, Statistical Methods in the Atmospheric Sciences, Elsevier, Oxford, 2006.

[71] Y. Kuramoto, T. Tsuzuki, On the formation of dissipative structures in reaction-diffusion systems: Reductive perturbation approach, Progr. Theoret. Phys. 54 (3) (1975) 687–699, http://dx.doi.org/10.1143/PTP.54.687.

[72] Y. Kuramoto, T. Tsuzuki, Persistent propagation of concentration waves in dissipative media far from thermal equilibrium, Progr. Theoret. Phys. 55 (2) (1976) 356–369, http://dx.doi.org/10.1143/PTP.55.356.

[73] G. Sivashinsky, Nonlinear analysis of hydrodynamic instability in laminar flames - I: Derivation of basic equations, Acta Astronaut. 4 (11) (1977) 1177–1206, http://dx.doi.org/10.1016/0094-5765(77)90096-0.

[74] G.I. Sivashinsky, D.M. Michelson, On irregular wavy flow of a liquid film down a vertical plane, Progr. Theoret. Phys. 63 (6) (1980) 2112–2114, http://dx.doi.org/10.1143/PTP.63.2112.

[75] R. Edson, J. Bunder, T. Mattner, A. Roberts, Lyapunov exponents of the Kuramoto–Sivashinsky PDE, ANZIAM J. 61 (2019) 270–285, http://dx.doi.org/10.21914/anziamj.v61i0.13939.

[76] R. Temam, Infinite-dimensional dynamical systems in mechanics and physics, second ed., in: Applied Mathematical Sciences, vol. 68, Springer-Verlag, New York, 1997, p. xxii+648, http://dx.doi.org/10.1007/978-1-4612-0645-3.

[77] E.N. Lorenz, Predictability: A problem partly solved, in: T. Palmer (Ed.), Proc. Seminar on Predictability, Vol. 1, ECMWF, Reading, UK, 1996, pp. 1–18.

[78] D.S. Wilks, Effects of stochastic parametrizations in the Lorenz '96 system, Q. J. R. Meteorol. Soc. 131 (606) (2005) 389–407, http://dx.doi.org/10.1256/qj.04.03.

[79] H.M. Arnold, I.M. Moroz, T.N. Palmer, Stochastic parametrizations and model uncertainty in the Lorenz 96 system, Phil. Trans. R. Soc. A 371 (1991) (2013) 20110479, http://dx.doi.org/10.1098/rsta.2011.0479.

[80] S. Herrera, J. Fernández, M. Rodriguez, J. Gutiérrez, Spatio-temporal error growth in the multi-scale Lorenz-96 model, Nonlinear Process. Geophys. 17 (4) (2010) 329, http://dx.doi.org/10.5194/npg-17-329-2010.

[81] 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, http://dx.doi.org/10.1016/j.jcp.2019.07.035.

[82] H. Kadri, E. Duflos, P. Preux, S. Canu, A. Rakotomamonjy, J. Audiffren, Operator-valued kernels for learning from functional response data, J. Mach. Learn. Res. 17 (20) (2016) 1–54, URL http://jmlr.org/papers/v17/11-315.html.