PROCEEDINGS A

royalsocietypublishing.org/journal/rspa

Simple, low-cost and accurate data-driven geophysical forecasting with learned kernels

B. Hamzi[1], R. Maulik[2] and H. Owhadi[3]

[1]Department of Mathematics, Imperial College London, London SW7 2AZ, UK

[2]Argonne Leadership Computing Facility, Argonne National Laboratory, Lemont, IL 60439, USA

[3]Department of Computing and Mathematical Sciences, Caltech, Pasadena, CA 91125, USA

RM, 0000-0001-9731-8936

Modelling geophysical processes as low-dimensional dynamical systems and regressing their vector field from data is a promising approach for learning emulators of such systems. We show that when the kernel of these emulators is also learned from data (using kernel flows, a variant of cross-validation), then the resulting data-driven models are not only faster than equation-based models but are easier to train than neural networks such as the long short-term memory neural network. In addition, they are also more accurate and predictive than the latter. When trained on geophysical observational data, for example the weekly averaged global sea-surface temperature, considerable gains are also observed by the proposed technique in comparison with classical partial differential equation-based models in terms of forecast computational cost and accuracy. When trained on publicly available re-analysis data for the daily temperature of the North American continent, we see significant improvements over classical baselines such as climatology and persistence-based forecast techniques. Although our experiments concern specific examples, the proposed approach is general, and our results support the viability of kernel methods (with learned kernels) for interpretable and computationally efficient geophysical forecasting for a large diversity of processes.

Research

Cite this article: Hamzi B, Maulik R, Owhadi H. 2021 Simple, low-cost and accurate data-driven geophysical forecasting with learned kernels. *Proc. R. Soc. A* **477**: 20210326.
https://doi.org/10.1098/rspa.2021.0326

Received: 28 April 2021
Accepted: 21 July 2021

Subject Areas:
atmospheric science, artificial intelligence, mathematical modelling

Keywords:
surrogate models, kernel methods, geophysical forecasting

Author for correspondence:
R. Maulik
e-mail: rmaulik@anl.gov

THE ROYAL SOCIETY
PUBLISHING

1. Introduction

The numerical simulation of geophysical processes for forecasting is limited by high computational costs, complex parametrization requirements and the spatio-temporal resolution of fast processes. Data-driven methods are becoming increasingly popular for surrogate modelling (emulation) or data analyses of all or a portion of the processes involved in numerical weather prediction [1–4]. These include deep learning and neural network methods [5–8], which have shown exceptional performance at the cost of a drop in interpretability. As an alternative, we combine interpretable kernel methods with kernel selection methods (kernel flows (KF) [9], as presented in [10] for learning dynamical systems) for the forecasting of geophysical processes using data and show competitive results at exceptionally low computational costs for sub-seasonal temperature forecasting when the underlying kernels are also learned from data. In particular, we demonstrate that these methods provide interpretable and computationally efficient alternatives[1] to deep learning architectures, which are infrastructure hungry and require off-nominal *post hoc* analyses for model interpretability such as Shapley additive explanations [14] or layerwise relevance propagation [15]. The framework proposed here is amenable to complex dynamics emulation, *a posteriori* error estimation and input and model-form uncertainty quantification while being computationally inexpensive from a learning perspective. However, we emphasize that the goal of this study is *not* to claim superiority over other geophysical emulators but to advocate the use of simple data-driven methods as effective baselines before migrating to compute hungry and poorly interpretable deep learning methods.

The rest of this document is organized as follows: §2 contains a literature review summarizing recent work in data-driven geophysical forecasting. Section 3 introduces the specific datasets that have been used for forecasting. The proposed algorithms for forecasting are introduced in §4. Section 5 outlines results from experiments and a discussion with concluding remarks is provided in §6.

2. Related work

Temperature forecasting is a crucial capability for several applications relevant to agriculture, energy, industry, tourism and the environment. Improved accuracy in short- and long-term forecasting of air and sea-surface temperature has significant implications for cost-effective energy policy, infrastructure development and downstream economic consequences [16]. The current state of the art in temperature forecasting is obtained with partial differential equation (PDE)-based methods [17,18]. Since these methods generally require solving large systems of equations with high-performance computing resources, they are limited by access and considerations of energy efficiency.

Therefore, the task of temperature (and, more broadly, geophysical process) forecasting has recently become a popular application of machine learning (ML) methods because of the promise of comparable (if not greater) forecast accuracy at a fraction of the computational cost. This also enables uncertainty quantification through ensemble forecasts [19], which are impossible for PDE-based methods owing to their excessive computational complexity. For these reasons, there has been a great degree of interest in building ML 'emulators' or 'surrogate models' from various geophysical datasets. There have been several studies on the use of ML for accelerating geophysical forecasts in recent times. Several rely on using ML methods to devise parametrizations for processes that contribute a high cost to the numerical simulation of the weather and climate [20–25]. In such cases, PDEs are not eschewed entirely. Other studies have looked at complete system emulators (i.e. forecasting from data alone) with a view to forecasting without any use of (and consequent limitations of) equation-based methods [6,26–31]. Other studies have looked at utilizing historical information for data-driven forecasting of specific

[1] Trained neural networks themselves can also be viewed as kernel machines [11,12] or warping kernel regressors [13].

processes [32–35] by focusing on specific influential variables or through the use of time-delay embeddings to offset inaccuracies due to unresolved variables. Further opportunities and perspectives for the use of data-driven methods for geosciences may be found in the reviews of [32,36]. Before proceeding, we note that a vast majority of the data-driven developments for geophysical forecasting have involved the use of variants of deep learning methods, for example ResNets [37], CapsuleNets [22], U-Nets [31], long short-term memory networks (LSTMs) [33], convolutional-LSTMs [34], neural ordinary differential equations (ODEs) [38,39] and local or global fully connected deep neural networks [32]. These methods, while exceptionally powerful in learning complex functions, hamper interpretability and require large computational resources for optimization. Furthermore, extensions to model form and data uncertainty quantification aggravate computational requirements significantly. Therefore, the goal of this research is to put forth viable alternatives to deep learning-based geophysical forecasting via the use of grey-box KF for learning dynamical systems.

In this article, we introduce an entirely data-driven method for forecasting the weekly averaged sea-surface temperature for the entire planet and the daily maximum air temperature over the North American continent. Furthermore, our method is developed to provide forecasts without the requirement for large-scale computational resources and with a greater degree of transparency. We achieve this by obtaining a low-dimensional affine subspace approximation of the temperature field on which a reduced system is evolved. Both dimensionality reduction and system evolution are performed using data-driven techniques alone, with the former employing a proper orthogonal decomposition (POD) and the latter combining kernel methods with a cross-validation technique known as KF. We remark that, in contrast to the growing popularity of deep learning methods for forecasting, we propose using classical dimensionality reduction and time-series forecasting with suitable inductive biases. Here, inductive biases are modifications to ML frameworks that 'assist' learning given prior knowledge of the physics they are employed to learn. Our competitive results motivate the creation of simple data-driven baselines that compare favourably with PDE-based methods without specialized neural architectures. At this point, we would like to emphasize to the reader that, while the experiments in this article devote themselves to forecasts on temperature, the proposed framework is general and can be applied to other geophysical flow-field forecasting, provided linear or nonlinear low-dimensional embeddings may be extracted, on which the dynamics evolve.

3. Dataset(s)

(a) Weekly averaged sea-surface temperature

For our first experiment, we use the open-source National Oceanic and Atmospheric Administration (NOAA) Optimum Interpolation sea-surface temperature V2 dataset (henceforth NOAA-SST).[2] This dataset has a strong periodic structure owing to seasonal fluctuations in addition to rich fine-scaled phenomena due to complex ocean dynamics. Weekly averaged NOAA-SST snapshots are available on a 0.25° grid which is sub-sampled to a 1° grid for the purpose of demonstrating our proposed methodology in a computationally efficient manner. This dataset, at the 1° resolution, has previously been used in several data-driven analysis tasks (for instance, see [40,41] for specific examples), particularly from the point of view of extracting seasonal and long-term trends as well as for flow-field recovery [42]. Each 'snapshot' of data originally corresponds to an array of size 360 × 180 (i.e. arranged according to the longitudes and latitudes of a 1° resolution). However, for effective utilization in forecasting, a mask is used to remove missing locations in the array that corresponds to the land area. Furthermore, it should be noted that forecasts are performed for those coordinates which correspond to oceanic regions

[2] Available at https://www.esrl.noaa.gov/psd/.

alone, and inland bodies of water are ignored. The non-zero data points then are subsequently flattened to obtain a column vector for each snapshot of our training and test data.

These data are available from 22 October 1981 to 30 June 2018 (i.e. 1914 snapshots for the weekly averaged temperature). We use the period of 22 October 1981 to 31 December 1989. The rest (i.e. 1990–2018) is used for testing. Our final number of snapshots for training amounts to 427, and for testing amounts to 1487. This train–test split of the dataset is a common configuration for data-driven studies [40] and the 8 year training period captures several short- and long-term trends in the global sea-surface temperature. Individual training samples are constructed by selecting a window of inputs (from the past) and a corresponding window of outputs (for the forecast task in the future) from the set of 427 training snapshots. From the perspective of notation, if $\boldsymbol{\theta}_t$ is a snapshot of training data, $t = 1, 2, \ldots, 427$, a forecasting technique may be devised by learning to predict $\boldsymbol{\theta}_{t+1}, \ldots, \boldsymbol{\theta}_{t+\tau}$ given $\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t-1}, \ldots, \boldsymbol{\theta}_{t-\tau}$. We note that this forecast is performed non-autoregressively—that is, the data-driven method is *not* utilized for predictions beyond the desired window size τ. Therefore, it is always assumed that the *true* $\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t-1}, \ldots, \boldsymbol{\theta}_{t-\tau}$ is available prior to making predictions. This means that, given a window of true inputs (for example obtained via an observation of the state of the system given re-analysis data), a forecast is made for a series of outputs (corresponding to the window length) before a metric of accuracy is computed for optimization. This is in contrast to a situation where simply one step of a prediction is used for computing a fitness metric which adversely affects the ability of the predictive method for longer forecasts into the future (owing to amplifying errors with each forecast step). The window size for the set of experiments on this dataset is fixed at eight weeks. The window-in and window-out construction of the training data leads to a final training dataset of size 411 samples. Since this dataset is produced by combining local and satellite temperature observations, it represents an attractive forecasting task for *non-intrusive* data-driven methods without requiring the physical modelling of underlying processes.

(b) North American daily midnight surface temperature

The National Centers for Environmental Prediction's (NCEP) North American Mesoscale Forecast System (NAM) [18] is one of the main mesoscale models used for guiding public and private sector meteorologists. NAM runs four times daily at three different spatial scales: (i) full North American 12 km resolution. (ii) Four kilometre continental US (CONUS) nest, 6 km Alaska nest and 3 km Hawaii and Puerto Rico nests. These domains are one-way nested inside the 12 km domain. (iii) High-resolution nested domain, which has a different location each cycle based upon the NCEP service centres and National Weather Service offices. For this work, analysis data from NAM were used, using the 12 km resolution grid for the surface temperature. The NAM data are collected in a time period between 28 October 2008 and 20 September 2018 on a daily cadence. In particular, we measure the temperature at midnight on each day in this temporal domain. This corresponds to 3569 snapshots of sea- and land-surface temperature data. In contrast to the previous problem, our goal is to forecast one week in advance with the temperature resolved daily. The first 2555 snapshots are reserved for the purpose of training the proposed forecasting technique, while the rest are used for testing. In a manner similar to the previous test case, a time delay of 7 days is used to specify the inputs to obtain the 7 day forecast of temperature in the future.

4. Methods

In this section, we shall introduce our proposed algorithm for forecasting. First, we seek to reduce the degrees of freedom of our forecast problem by performing a dimensionality reduction using the POD. This reduced representation of our dataset is then used to train an efficient dynamical systems emulator using kernel methods. Forecasts from this emulator may then be reconstructed in the original space through the use of previously computed POD basis functions. The overall schematic of this procedure is shown in figure 1.

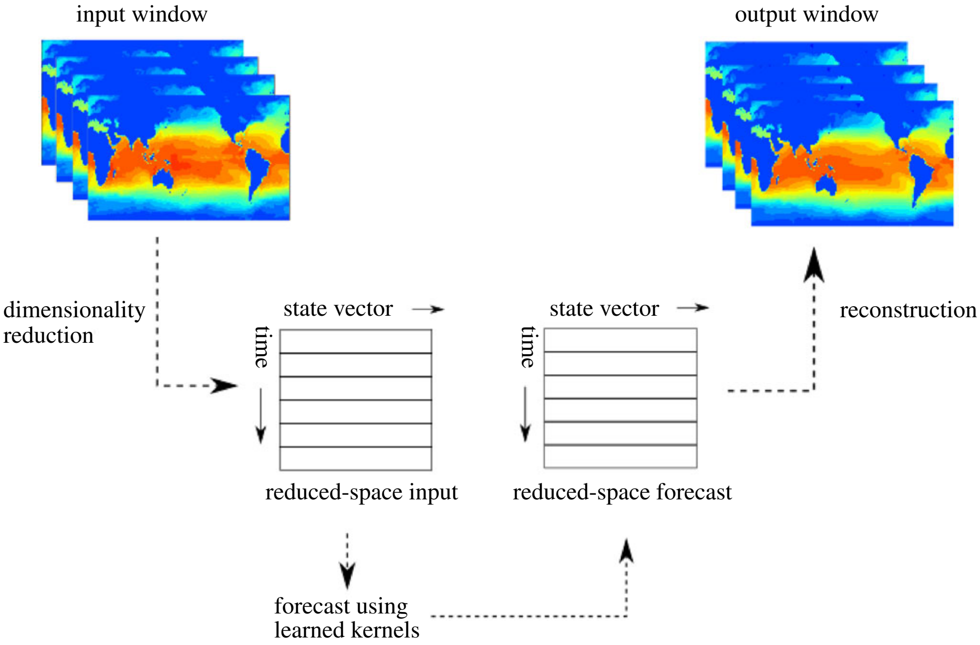

Figure 1. The overall schematic of the proposed workflow. Learned kernels are used to forecast the state of a geophysical process on a low-dimensional embedding obtained by the POD. Forecasted trajectories in the reduced subspace are reconstructed to obtain the final predicted state. (Online version in colour.)

(a) Dimensionality reduction: proper orthogonal decomposition

POD provides a systematic method to project dynamics of a high-dimensional system onto a lower-dimensional subspace. We suppose that a single snapshot of the full system is a vector in $\mathbb{R}^N$, where N could be the number of grid points at which the field is resolved. Observing the system across a number of time points gives us the snapshots $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_T$, with the mean subtracted by convention. The aim of POD is to find a small set of orthonormal basis vectors $\boldsymbol{v}_{\mathbf{1}}, \dots, \boldsymbol{v}_M$, with $M \ll N$, which approximates the spatial snapshots,

$$\boldsymbol{\theta}_t \approx \sum_{j=1}^{M} a_j(t)\boldsymbol{v}_j, \quad t = 1, \dots, T, \tag{4.1}$$

and so allows us to approximate the evolution of the full N-dimensional system by considering only the evolution of the M coefficients $a_j(t)$. POD chooses the basis, $\boldsymbol{v}_j$, to minimize the residual with respect to the L_2 norm,

$$R = \sum_{t=1}^{T} \left\| \boldsymbol{\theta}_t - \sum_{j=1}^{M} a_j(t)\boldsymbol{v}_j \right\|^2 . \tag{4.2}$$

Defining the snapshot matrix, $\boldsymbol{S} = [\boldsymbol{\theta}_1 | , \dots, |\boldsymbol{\theta}_T]$, the optimal basis is given by the M eigenvectors of $\boldsymbol{S}\boldsymbol{S}^T$, with largest eigenvalues, after which the coefficients are found by orthogonal projection, $\mathbf{a}(t) = \langle \boldsymbol{\theta}_t, \boldsymbol{v} \rangle$ [43]. The coefficients $\mathbf{a}(t)$ correspond to a time series with an M-dimensional state vector, which is the focus of our temporal forecasting.

For both of our datasets, we take only the training data snapshots, say $\boldsymbol{D}_1, \dots, \boldsymbol{D}_T$, from which we calculate the mean $\bar{\boldsymbol{D}} = (1/T) \sum_t \boldsymbol{D}_t$, hence defining the mean subtracted snapshots $\boldsymbol{\theta}_t = \boldsymbol{D}_t - \bar{\boldsymbol{D}}$. We then create the snapshot matrix, $\boldsymbol{S}$, and find numerically the M eigenvectors of $\boldsymbol{S}\boldsymbol{S}^T$ with largest eigenvalues. From this, we train models, $f^\dagger$ (to be defined in the next section), to forecast the coefficients

$$(\boldsymbol{a}(t+1), \boldsymbol{a}(t+2), \ldots, \boldsymbol{a}(t+\tau)) \approx (\hat{\boldsymbol{a}}(t+1), \hat{\boldsymbol{a}}(t+2), \ldots, \hat{\boldsymbol{a}}(t+\tau))$$
$$= f^{\dagger}(\boldsymbol{a}(t), \boldsymbol{a}(t-1), \ldots, \boldsymbol{a}(t-\tau)), \tag{4.3}$$

making predictions of future coefficients given previous ones. Here, τ corresponds to the time delay that is embedded into the input feature space for the purpose of forecasting. In this article, τ also stands for the forecast length obtained from the fit model, although, in the general case, these quantities can be chosen to be different. We can now test for predictions on unseen data, $\boldsymbol{E}_1, \ldots, \boldsymbol{E}_Q$, where $\boldsymbol{E}_t$ is an unseen snapshot of data at time t and Q is the total number of test snapshots. Note that these test snapshots are obtained for a time interval that has not been used for constructing the POD basis vectors. We proceed by taking the mean $\bar{\boldsymbol{D}}$ and vectors $\boldsymbol{v}_j$ calculated from the training data to get test coefficients at an instant in time

$$a_j(t) = \langle \boldsymbol{E}_t - \bar{\boldsymbol{D}}, \boldsymbol{v}_j \rangle, \quad j = 1, \ldots, M, \tag{4.4}$$

which will be used with the model $f^{\dagger}$ to make future predictions in the unseen testing interval. The prediction for the coefficients $\hat{\boldsymbol{a}}$ can be converted into predictions in the physical space by taking $\bar{\boldsymbol{D}} + \sum_j \hat{a}_j \boldsymbol{v}_j$. This procedure only makes use of testing data to pass into the model, not to train the model in any way. Crucially, to make a forecast of $\boldsymbol{E}_{t+1}, \boldsymbol{E}_{t+2}, \ldots, \boldsymbol{E}_{t+\tau}$, only previous measurements $\boldsymbol{E}_t, \boldsymbol{E}_{t-1}, \ldots, \boldsymbol{E}_{t-\tau}$ are needed.

(b) Cross-validation for temporal forecasting

The method employed for temporal forecasting was introduced in [10] for learning dynamical systems from data. It is a kernel regression with a kernel learned from data from a variant of cross-validation known as KF [9].

To describe this method consider the problem of forecasting $\boldsymbol{a}(n+1)$ given the observation of $\boldsymbol{a}(1), \ldots, \boldsymbol{a}(n)$, where $\boldsymbol{a}(1), \ldots, \boldsymbol{a}(k), \ldots$ is a time series in $\mathbb{R}^M$ (in our application, M is the number of modes, used for prediction). Evidently making a forecast requires some assumptions on the time series and the assumption that we work under is that it can be approximated by a solution of a dynamical system of the form

$$z_{n+1} = f^{\dagger}(z_n, \ldots, z_{n-\tau^{\dagger}+1}), \tag{4.5}$$

where the time delay $\tau^{\dagger} \in \mathbb{N}^{+}$ (the positive natural numbers) and $f^{\dagger}$ is the vector field, which may be unknown. Here, in our application, each z_n is a time window of POD coefficients, i.e. of the form $z_n = (\boldsymbol{a}(n-\tau), \ldots, \boldsymbol{a}(n))$, where τ is the length of the input and forecast windows ($\tau = 8, 7$ in our respective datasets). A simple solution to our extrapolation problem is then, given $\tau \in \mathbb{N}^{+}$, to learn/approximate $f^{\dagger}$ from data (the past values of the time series), and then use the approximate vector field f to define a surrogate model

$$z_{n+1} = f(z_n, \ldots, z_{n-\tau+1}),$$

for predicting the future states of the dynamical system. Therefore, the approximation of the dynamical system can be recast as that of regressing/interpolating $f^{\dagger}$ from pointwise measurements

$$f^{\dagger}(X_n) = Y_n \quad \text{for } n = 1, \ldots, N - \tau, \tag{4.6}$$

with $X_n := (\boldsymbol{a}(n+\tau-1), \ldots, \boldsymbol{a}(n))$ and $Y_k := \boldsymbol{a}(n+\tau)$. For the experiments in this study, forecasts of length τ are concatenated during testing to obtain predictions in the entire testing time scales. Given a kernel K, this interpolant (in the absence of measurement noise) is

$$f(x) = K(x, X)(K(X, X))^{-1} Y, \tag{4.7}$$

where $X = (X_1, \ldots, X_{N-\tau})$, $Y = (Y_1, \ldots, Y_{N-\tau})$, $K(X, X)$ stands for the $N - \tau \times N - \tau$ matrix with entries $K(X_i, X_j)$ and $K(x, X)$ is the $N - \tau$ vector with entries $K(x, X_i)$. Writing ξ for the centred Gaussian process with covariance function K and $\mathcal{H}$ for the reproducing kernel Hilbert spaces

(RKHS) defined by K (see appendix A for a reminder), recall that (4.7) is (i) the conditional expectation of ξ given the measurements $\xi(X_i) = Y_i$ and (ii) a minimax optimal approximation [44] of $f^{\dagger}$ in $\mathcal{H}$ (using the relative error in the RKHS norm $||\cdot||_{\mathcal{H}}$ as a loss).

Evidently, the implementation of the proposed approach requires the prior selection of a kernel K and here we propose (as in [10]) to also learn that kernel K from data using a variant of cross-validation known as KF [9]. Given a family of kernels $K_\theta(x,x')$ parametrized by θ, the KF algorithm seeks to select a θ such that subsampling the data does not influence the interpolant much. Writing $||\cdot||_{K_\theta}$ for the RKHS norm defined by the kernel $K_\theta(x,x')$, and two interpolants u^b and u^c obtained with the kernel K_θ and by subsampling the data (u^c uses a smaller subset of the data than u^b), the KF algorithm seeks to learn θ by successively moving θ in the gradient descent direction of the loss $\rho = ||u^b - u^c||^2_{K_\theta}/||u^b||^2_{K_\theta}$ (4.8). Note that (i) ρ is a cross-validation term (randomized through the subsampling procedure) acting as a surrogate for the generalization error (in relative RKHS norm) and (ii) u^b acts as a surrogate for the target function. To summarize, given a family of kernels $K_\theta(x,x')$ parametrized by θ, the KF algorithm is as follows [9,45]:

(i) Randomly select subvectors X^b and Y^b of X and Y (through uniform sampling without replacement in the index set $\{1,\ldots,N-\tau\}$).

(ii) Randomly select subvectors X^c and Y^c of X^b and Y^b (by selecting, at random, uniformly and without replacement, half of the indices defining X^b).

(iii) Let[3]

$$\rho(\theta, X^b, Y^b, X^c, Y^c) := 1 - \frac{Y^{c,T} K_\theta(X^c, X^c)^{-1} Y_c}{Y^{f,T} K_\theta(X^b, X^b)^{-1} Y^b} \tag{4.8}$$

be the squared relative error (in the RKHS norm $||\cdot||_{K_\theta}$ defined by K_θ) between the interpolants u^b and u^c obtained from the two nested subsets of the dataset and the kernel K_θ.

(iv) Evolve θ in the gradient descent direction of ρ, i.e. $\theta \leftarrow \theta - \delta\nabla_\theta \rho$.

(v) Repeat until desired accuracy is achieved or computational budget is exhausted.

A schematic of the kernel learning process is shown in figure 2. In this study, we use the following expression for our kernel:

$$\begin{aligned} K_\theta(x,x') = {} & \theta_1^2 \exp\left(-\frac{||x-x'||^2}{2\theta_2^2}\right) + \theta_3^2\left(1+\theta_4^2 x^T x'\right)^2 + \theta_5^2\left(1+\theta_6^2||x-x'||^2\right)^{-(1/2)} \\ & + \theta_7^2\left(1+\theta_8^2||x-x'||^2\right)^{\theta_9} + \theta_{10}^2\left(1+\theta_{11}^2||x-x'||^2\right)^{-1} \\ & + \theta_{12}^2 \max\left(0, 1-\theta_{13} * ||x-x'||^2\right). \end{aligned} \tag{4.9}$$

The main motivation for this kernel is the numerical experiments in [10], where the authors used KF to learn prototypical chaotic dynamical systems that exhibit a rich dynamic behaviour. The motivation of the triangular kernel is that the Bernoulli map that exhibits a rich dynamic behaviour is toplogically conjugate to the unit-height tent map. The quadratic kernel captures the long-term correlation and the Gaussian kernel captures the short-term correlation. Our experiments suggest that the particular parametrized family of kernels is not important as long as it is rich enough (to reproduce the patterns contained in the dataset). The additive form of the kernel is equivalent to regressing the data with a sum of independent Gaussian processes, each one being adapted to a particular feature/pattern of the data. As presented in [46], additive kernels can be employed to program sophisticated kernels for pattern recognition tasks.

In the following, we denote the use of cross-validation to forecast POD coefficients as the *POD-RKHS* emulation framework.

[3] $\rho := ||u^b - u^c||^2_{K_\theta}/||u^b||^2_{K_\theta}$, with $u^b(x) = K_\theta(x, X^b)K_\theta(X^b, X^b)^{-1}Y^b$ and $u^c(x) = K_\theta(x, X^c)K_\theta(X^c, X^c)^{-1}Y^c$, and ρ admits the representation (4.8), enabling its computation.

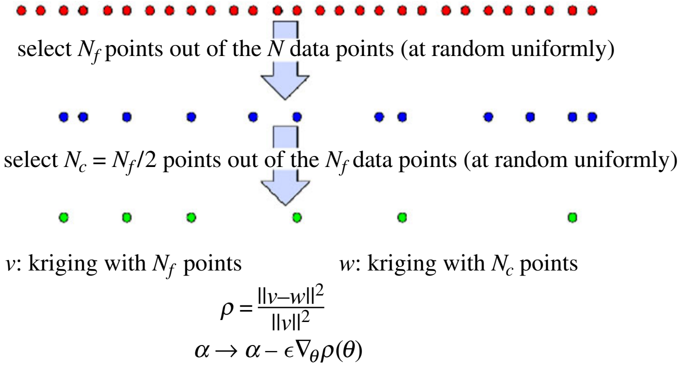
select N_f points out of the N data points (at random uniformly)
select $N_c = N_f/2$ points out of the N_f data points (at random uniformly)
v: kriging with N_f points
w: kriging with N_c points
$\rho = \frac{\|v-w\|^2}{\|v\|^2}$
$\alpha \to \alpha - \epsilon \nabla_\theta \rho(\theta)$

Figure 2. Learning kernel hyperparameters, θ, by cross-validation from data. A workflow schematic. (Online version in colour.)

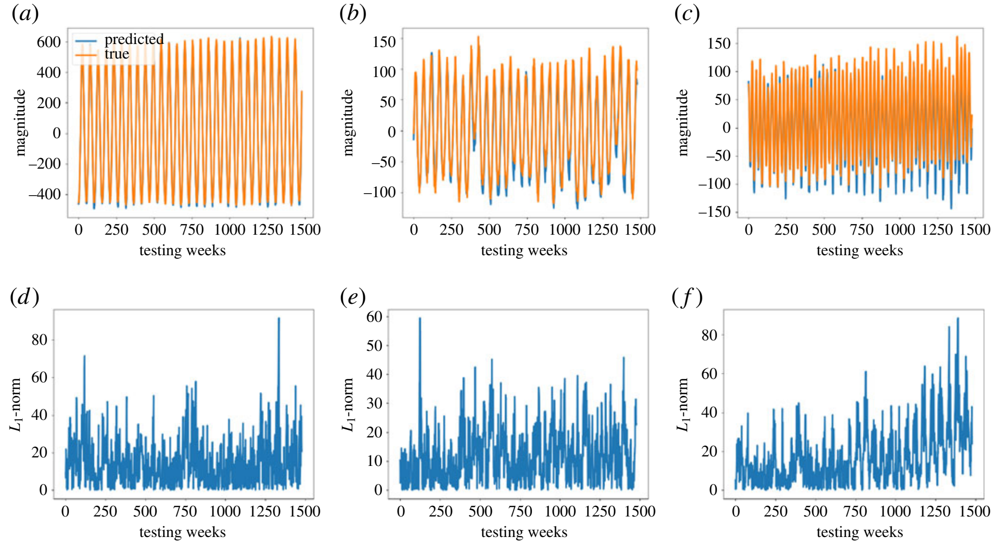

Figure 3. POD coefficient predictions (*a*–*c*) and L_1 errors with corresponding true values (*d*–*f*) for the NOAA-SST dataset using POD-RKHS. We use 427 snapshots between 22 October 1981 and 31 December 1989 for training. The remaining 1487 snapshots of the dataset (i.e. 1990–2018) are used for testing and are shown here for model assessment. (*a*) Coefficient 1, (*b*) coefficient 2, (*c*) coefficient 3, (*d*) coefficient 1 error, (*e*) coefficient 2 error, (*f*) coefficient 3 error. (Online version in colour.)

5. Experiments

In this section, we outline results from the use of the POD-RKHS framework for forecasting on the aforementioned datasets. We also provide metrics that evaluate the accuracy of the forecasts and compare them with baseline techniques. Our first set of results are shown for the NOAA-SST dataset. The ability to forecast weekly averaged coefficients eight weeks at a time is shown in figure 3 (for the testing period), where it can be observed that the lower order structures are predicted with high fidelity. The framework shows deviations in the finer scale content (mode 3) as one approaches the end of the testing period, indicating potential extrapolation. Similar behaviour was observed in [33,35] as well, where a multi-cell LSTM was used to forecast in this reduced space (henceforth POD-LSTM). Time-series assessments for various point probes in the Eastern Pacific are shown for a testing sub-window where data from all prediction sources were available (between 5 April 2015 and 17 June 2018) in figure 4. The plot indicates a competitive testing performance when compared with state-of-the-art equation-based methods

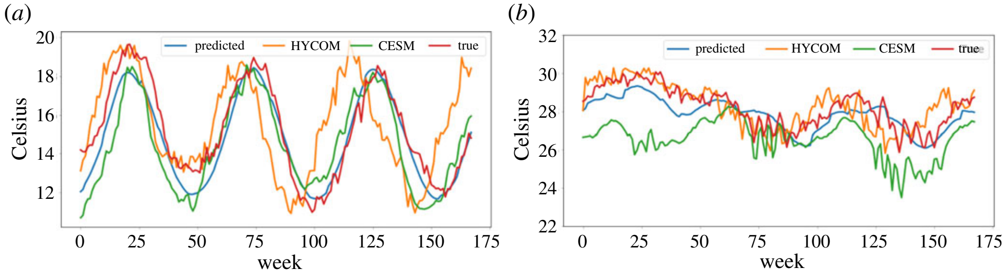

Figure 4. Probe time-series comparisons for NOAA-SST from CESM, HYCOM and POD-RKHS within the testing regime at two different locations. The data are plotted for the weeks between 5 April 2015 and 17 June 2018. (*a*) 50° latitude, 230° longitude and (*b*) 85° latitude, 210° longitude. (Online version in colour.)

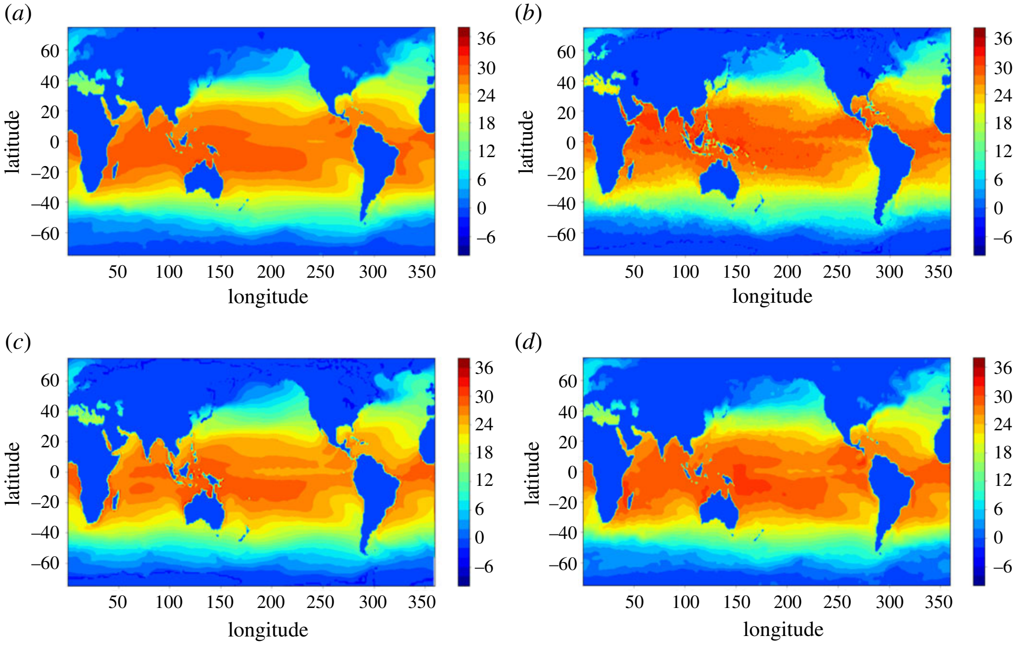

Figure 5. Contour plots for various forecasts ((*a*) POD-RKHS, (*b*) HYCOM, (*c*) CESM, (*d*) truth), indicating how low-dimensional manifold-based emulation reduces the ability to capture fine-scaled features in the flow-field. HYCOM, with the finest resolution, is seen to capture small-scale information most accurately. Note, however, that the POD-based emulation framework is competitive in an averaged sense, as seen through RMSE metrics. (Online version in colour.)

such as the Community Earth System Model (CESM) [47] and the US Navy Hybrid Coordinate Ocean Model (HYCOM) [48]. The former is a climate simulator and the latter is an operational forecast system giving short-term predictions. Both these techniques require vast computational resources, whereas our forecast is performed on a single-node machine without any accelerator.

Some qualitative comparisons of the predictions are shown in figure 5, where an acceptable agreement between different methods and the remote-sensing dataset is observed (table 1). We also note that the metrics obtained using the optimized LSTM architecture in this table outperformed the accuracy obtained from classical linear or decision-tree-based forecasting techniques (see table 2 in [33]). A closer examination of the mean squared error during the entire testing period is shown in figure 6, where POD-RKHS is seen to give sufficiently accurate predictions in the entire domain, including the vital Eastern Pacific region. With the testing period

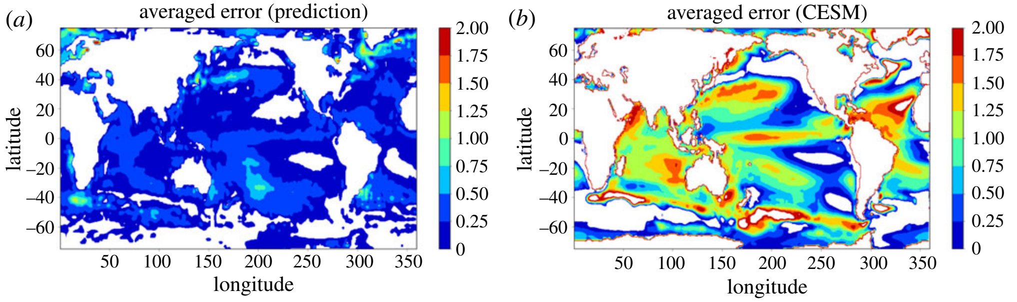

Figure 6. Contour plots for RMSE over the entire testing period (1990–2018) for the NOAA-SST dataset for POD-RKHS (*a*) and CESM (*b*). The results indicate that the proposed low-dimensional emulator captures the long-term behaviour of the SST fluctuations more accurately than CESM. (Online version in colour.)

Table 1. RMSEs (in Celsius) for different forecast techniques compared against the POD-RKHS forecasts between 5 April 2015 and 24 June 2018, in the Eastern Pacific region (within –10° to +10° latitude and 200–250° longitude). POD-RKHS matches the accuracy of the process-based models for this particular metric and assessment and that of an optimized LSTM [33] using neural architecture search.

	RMSE (°)							
	week 1	week 2	week 3	week 4	week 5	week 6	week 7	week 8
POD-LSTM	0.62	0.63	0.64	0.66	0.63	0.66	0.69	0.65
CESM	1.88	1.87	1.83	1.85	1.86	1.87	1.86	1.83
HYCOM	0.99	0.99	1.03	1.04	1.02	1.05	1.03	1.05
POD-RKHS	0.76	0.67	0.66	0.69	0.69	0.72	0.77	0.76

being around 18 years, the El Niño–Southern Oscillation (ENSO), occurring at a frequency of 2–7 years [4], is captured more accurately in comparison with CESM.

In terms of times to solution, data-driven models provided instantaneous forecasts for the given time period (1981–2018). By contrast, equation-based models would require larger computing resources and times to solution by several orders of magnitude. For example, CESM (for the forecast period of 1920–2100) required 17 million core-hours on Yellowstone, the National Center for Atmospheric Research (NCAR) high-performance computing resource, for obtaining forecasts for each member of a 30-member ensemble. While finer details about computational costs were not readily available for HYCOM, this short-term ocean prediction system runs daily at the Navy Department of Defense Supercomputing Resource Center, with data typically accessible within 48 h of the simulation initialization.[4] Benchmarking results for the 1/25° HYCOM forecasts (much finer than the reference data used here) indicate the requirement of 800 core-hours per day of forecast on a Cray XC40 system.[5] For a more appropriate comparison, we also take into account the computational cost of the search for an optimal LSTM architecture and its training. While the POD-RKHS method requires approximately 40 s to train on a single-node machine without acceleration, the LSTM being compared with was discovered using 3 h of wall time on 128 compute nodes of the Theta supercomputer with the Intel Knights Landing architecture. Note that POD-RKHS was trained only once on one Intel CoreI7 X86_64 CPU without any hardware acceleration. Subsequent assessments on a different dataset (in the next set of experiments) indicate that the RKHS continues to be competitive in training costs. All experiments involving

[4] https://www.hycom.org/dataserver.

[5] https://www.hycom.org/attachments/066_talk_COAPS_17a.pdf.

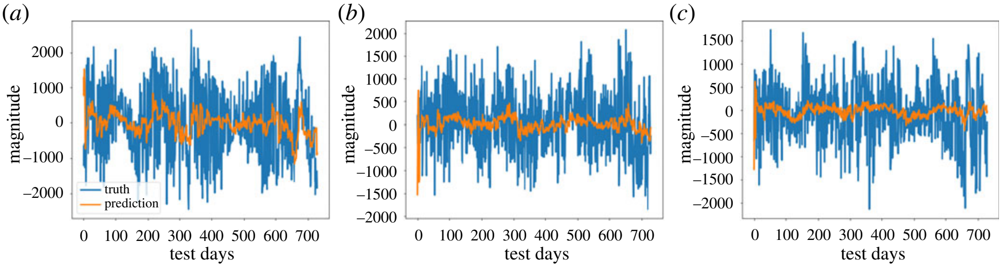

Figure 7. POD coefficient predictions for forecasting in the testing regime for the NOAA-NCEP NAM dataset. The plots indicate the stochastic nature of the daily temperature at midnight and also highlight the ability of the POD-RKHS framework to extract a signal from them. (*a*) Coefficient 1, (*b*) coefficient 2, (*c*) coefficient 3. (Online version in colour.)

POD-RKHS were performed using Python with the NumPy and AutoGrad libraries being used for numerical linear algebra and optimization, respectively.

Our second set of assessments are performed for the NOAA-NCEP NAM dataset for sea- and land-surface air temperatures over the North American continent. This dataset is obtained by a re-analysis of numerical simulation data obtained through data assimilation of observations. We follow the same strategy of time-delayed inputs to outputs for forecasting the midnight temperature. Figure 7 shows the ability of the time-series emulator to extract an underlying trend from a highly noisy signal.

Note that the difficulty of extracting the extremely high-frequency features contributes to the poor capture of extreme events during emulation. The reconstruction accuracy from the forecast is compared in a series of assessments beginning with root-mean-squared error (RMSE) assessments as shown in figure 8 for the testing time period. POD-RKHS is seen to provide competitive results in comparison with persistence and climatology at the lower latitudes. Here, climatology refers to the temperature on a particular day averaged over multiple years, and persistence is the last known (i.e. last observed) temperature on the day the forecast is made. By contrast, exceptional gains are seen at the higher latitudes with far reduced RMSE values from the proposed framework. Note that errors across all frameworks were maximum in the northern section of the continent, proving the increasing difficulty of predicting in that region using data-driven techniques. Figure 9 plots the contours for improvements in the correlation coefficient and cosine similarities[6] of the predictions when compared with climatology. While correlation coefficients indicate regions where the predictions improve over the baselines in an averaged sense, cosine similarities indicate the ability to detect extreme fluctuations in a forecast. The contours indicate that the proposed framework improves on extreme fluctuation detection over climatology in the northern region of the spatial domain as well. Similar trends are observed for comparisons with persistence with large gains in cosine similarities in the northern part of the spatial domain.

For the purpose of comparison, we also outline the performance of a standard LSTM framework (manually selected in comparison with [33]) for forecasting on the NAM dataset. The problem formulation is identical, with the LSTM using a window of inputs of 7 days length and forecasting the POD coefficient trajectories for the next 7 days. We train 55 instances of this LSTM, consisting of three stacked LSTMs and 50 neurons for each of the linear operations in each cell, and select the one with the best training and validation performance. The 55 LSTM architectures were trained using an Nvidia V100 GPU and required 5 h of wall time. For each training, an early stopping criterion is used to terminate training if validation losses do not improve for 10 successive training epochs. Figure 10 shows predictions from the testing data using the trained LSTM with clear indications of poor capture of dynamics. These are confirmed by contour plots for the RMSE, shown in figure 8, where the POD-LSTM framework is the weakest

[6] The cosine similarity between two vectors $\mathbf{a}$ and $\mathbf{b}$ is $(\mathbf{a}\cdot\mathbf{b})/||\mathbf{a}||||\mathbf{b}||$.

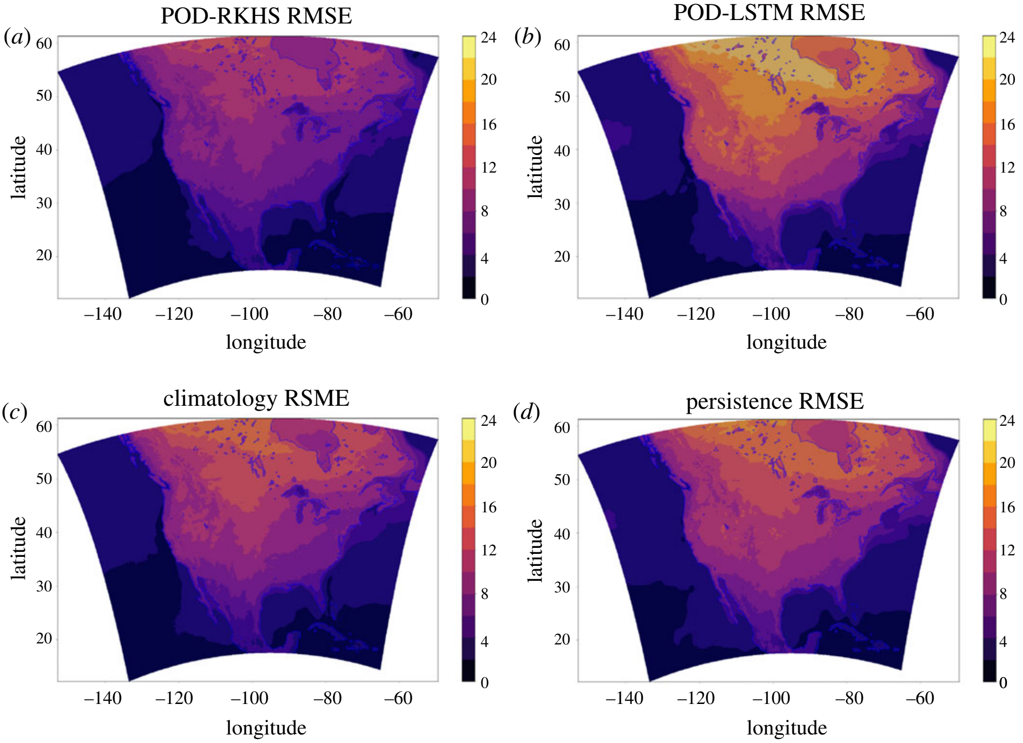

Figure 8. Testing RMSEs for predictions on the NAM data as compared with (*c*) climatology and (*d*) persistence baselines for the (*a*) POD-RKHS and (*b*) POD-LSTM frameworks. This result indicates the use of one potential LSTM with manual tuning—further improvements may be obtained through neural architecture and a hyperparameter search as seen in [33]. (Online version in colour.)

of all the data-driven methodologies presented here. Figure 11 examines potential improvements in the correlations and cosine similarities within the spatial domain of the dataset for this experiment. Climatology and persistence are seen to comprehensively outperform a standard LSTM architecture. Note that, as suggested by figures 3 and 7, most of the forecast error is due to bias for the NOAA-SST dataset (low-noise dataset) and variance for the NOAA-NCEP NAM dataset (high-noise dataset). It must be noted that this result is not meant to justify mistrust of all deep learning architectures but to propose the use of classical kernel-based methods for the construction of baselines.

We now compare the training and testing times of the proposed framework with those of LSTM methods with standard architectures. The POD-RKHS method required a total of 136 s to train, whereas the POD-LSTM methods took an average of 330 s to train per LSTM architecture. Although the choice of user-defined hyperparameters in both these methods may change these values, their training times are comparable. We note that, in a manner similar to [33], discovering the optimal hyperparameters and architectures for a deep learning framework adds significant costs associated with sampling a high-dimensional search space of neural networks that may fit the given training data. In that regard, kernel methods may provide significant computational gains.

6. Discussion

In this article, we have employed a simple kernel regression method for the forecasting of geophysical time series. We have shown that when the kernel is also learned from data (via KF), then the proposed framework provides competitive and computationally efficient baselines for

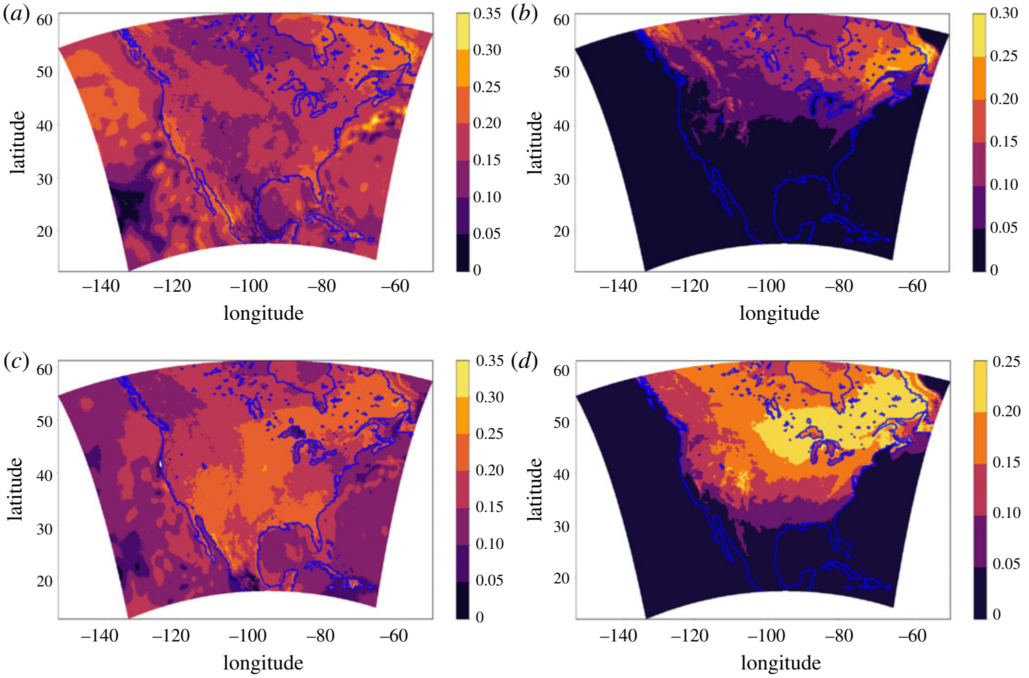

Figure 9. Metric *improvements* in climatology (*a*,*b*) and persistence (*c*,*d*) using the POD-RKHS time-series emulator. Lighter regions depict areas where the POD-LSTM forecast model outperformed the climatology and persistence baselines. The north domain of the dataset shows improved performance by the proposed framework, particularly for extreme fluctuation detection (indicated by cosine similarity improvement). Overall correlation with the truth is seen to be improved throughout the domain. (*a*,*c*) Correlation coefficients and (*b*,*d*) cosine similarities. (Online version in colour.)

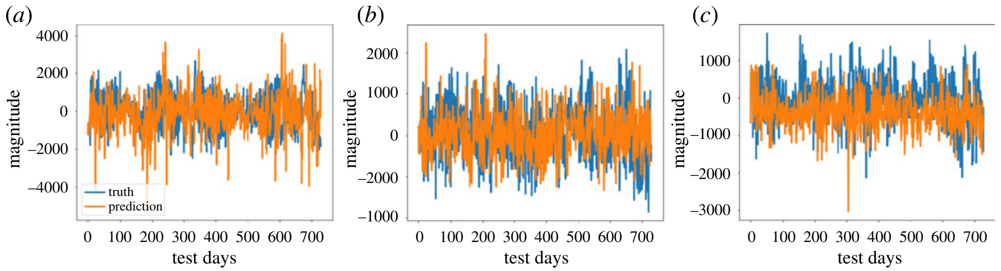

Figure 10. Predictions for the NOAA-NCEP NAM POD coefficients using a POD-LSTM framework. This prediction was obtained with three stacked LSTM cells, each with 50 neurons for all the internal operations. An LSTM that provided the best training and validation performance among 55 restarts was chosen. The total time to train these LSTMs was 5 h on an Nvidia V100 GPU. It is observed that the LSTM architecture is sensitive to the stochastic nature of the data. (*a*) Coefficient 1, (*b*) coefficient 2, (*c*) coefficient 3. (Online version in colour.)

the geophysical emulation of two temperature-based datasets given by the weekly averaged sea-surface temperature (NOAA-SST) and the daily resolved midnight temperature (NCEP-NAM). In both cases, the proposed method recovers a stable temporal behaviour on a low-dimensional manifold. Importantly, for the second dataset, strong stochasticity is handled robustly by the proposed framework to obtain superior metrics when compared with classical baselines such as climatology or persistence. Comparisons with the POD-LSTM framework, commonly used for non-intrusive reduced-order modelling, are also favourable. While the POD-LSTM obtained through an expensive parallelized neural architecture search slightly outperforms the proposed

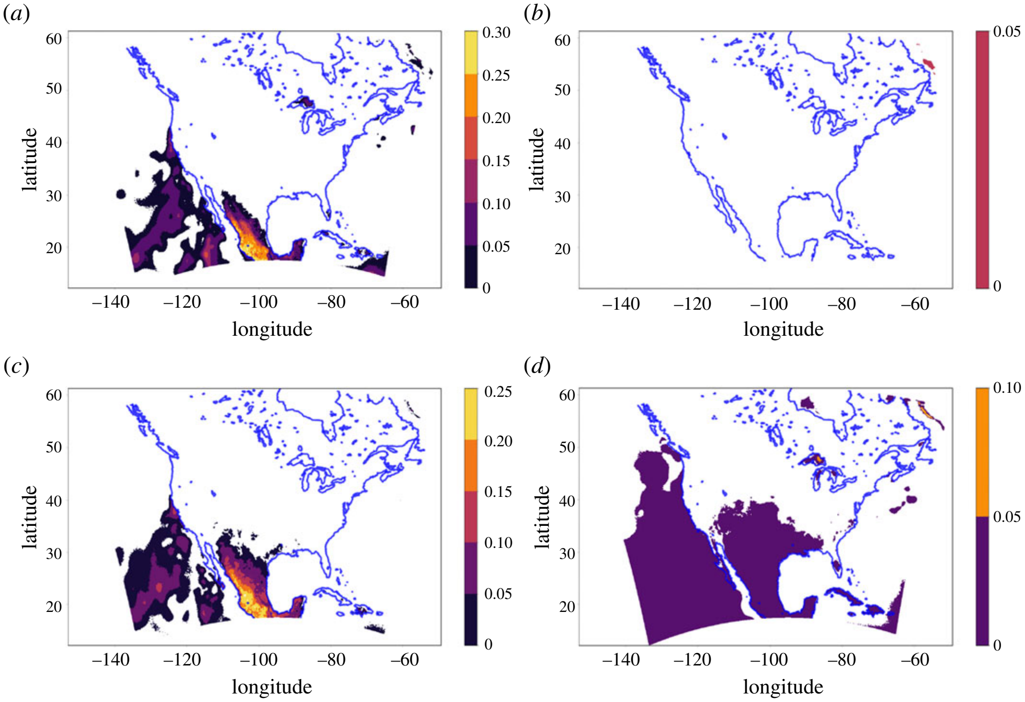

Figure 11. Metric *improvements* on climatology (*a*,*b*) and persistence (*c*,*d*) using the LSTM time-series emulator. Lighter regions depict areas where the POD-LSTM forecast model outperformed the climatology and persistence baselines. Vast stretches (i.e. the white regions) of the domain show poorer performance for POD-LSTM against climatology and persistence baselines. (*a*,*c*) Correlation coefficients and (*b*,*d*) cosine similarities. (Online version in colour.)

framework for the relatively smooth NOAA-SST time-series data, the proposed framework is much more successful in the case of the stochastic NCEP-NAM temperature data. The computational cost of learning kernels in the proposed method is considerably lower than PDE-based forecasting alternatives and does not require any specialized hardware (such as artificial intelligence accelerators). When considering the fact that deep learning frameworks often require significant architecture tuning to obtain a viable function approximation, our framework provides greater computational gains than those data-driven surrogates over the lifetime of a forecast campaign.

Extensions to this work include the addition of more predictive variables to the training data to emulate larger fractions of the Earth system model. This may be obtained by concatenating reduced representations of a greater number of flow-field variables (in truncated POD space) to obtain a larger, more informative state vector. We note that, despite the successes of the framework in this article, several forecast tasks may be challenging using the POD-RKHS methodology. These would generally be due to the absence of a low-dimensional manifold on which a large portion of the flow-field dynamics is restricted. An example forecast task would be something akin to the prediction of precipitation. Therefore, future extensions must account for situations where a linear compression, given by the POD, is inefficient because of the presence of very fine scales in the data—in such cases, nonlinear compression techniques such as diffusion maps or neural network parametrized autoencoders may be necessary for effective dimensionality reduction. Here, an interesting avenue for exploration is to use KF for obtaining reduced representations themselves in a unified pipeline for geophysical emulation. These directions may be instrumental for pushing data-driven forecast horizons to regimes where classical methods are untrustworthy. Finally, a major advantage of using KF is that posterior distributions may be learned under the Gaussian assumption, which would allow for aleatoric uncertainty quantification. A further assumption of hyperparameters being random variables that are sampled from a joint distribution would enable epistemic uncertainty estimates as well. These ideas are our current research focus.

Data accessibility. The data that support the findings of this study are openly available in Github at https://github.com/Romit-Maulik/POD_RKHS.

Authors' contributions. B.H. prepared the code and analysis and wrote portions of the paper. R.M. designed the investigation, prepared code and documentation, performed analyses, generated visualizations and wrote the paper. H.O. prepared code and analysis and wrote portions of the paper.

Competing interests. We declare we have no competing interests.

Funding. This material is based upon work supported by the US Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, under contract DE-AC02-06CH11357. This research was funded in part by and used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under contract DE-AC02-06CH11357. R.M. acknowledges support from the Margaret Butler Fellowship at the Argonne Leadership Computing Facility. B.H. thanks the European Commission for funding through the Marie Curie fellowship STALDYS-792919 (Statistical Learning for Dynamical Systems). H.O. gratefully acknowledges support by the Air Force Office of Scientific Research under award nos. FA9550-18-1-0271 (Games for Computation and Learning) and MURI (FA9550-20-1-0358).

Disclaimer. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the US DOE or the US government.

Appendix A. Gaussian process regression

The interpolant (4.7) can also be identified as the conditional mean of the centred GP $\xi \sim \mathcal{N}(0, K)$ with K as the covariance function conditioned on $\xi(X_i) = Y_i$, Furthermore, we have the pointwise error estimate (see [49] and [50, theorem 5.1])

$$|f^\dagger(x) - f(x)| \leq \sigma(x) ||f^\dagger||_K, \tag{A 1}$$

bounding interpolation error between $f^\dagger$ and f, where $||f^\dagger||_K$ is the RKHS norm defined by the kernel K and

$$\sigma^2(x) = K(x,x) - K(x,X)(K(X,X))^{-1}K(x,X)^T \tag{A 2}$$

is the conditional variance of the GP ξ. From a practical point of view $||f^\dagger||_K$ could be replaced by the RKHS norm of its interpolant $\sqrt{Y^T(K(X,X))^{-1}Y}$ to offer a rough error estimate.[7]

Appendix B. Error estimates

For the sake of completeness, we include convergence results from [51,52] that characterize the error estimates of the difference between a dynamical system defined by an ODE and its approximation from data.

We consider ODEs of the form

$$\dot{x} = f^*(x), \tag{B 1}$$

where $f^* : \mathbb{R}^d \to \mathbb{R}^d$ is a smooth vector field and dot denotes differentiation with respect to time.

We assume that the function f^* is unknown, but we have sampled data of the form (x_i, y_i) in $X \times \mathbb{R}^d$, $i = 1, \ldots, m$, with

$$y_i = f^*(x_i) + \eta_{x_i}. \tag{B 2}$$

We assume that the one-dimensional random variables $\eta^k_{x_i} \in \mathbb{R}^d$, where $i = 1, \ldots, m$ and $k = 1, \ldots, d$, are independent random variables drawn from a probability distribution with zero mean and variance $(\sigma^k_{x_i})^2$ bounded by σ^2.

The function spaces that we use to search for our approximations to both f^* will be RKHS. For a survey of the main properties of RKHS spaces mentioned in this section, we refer to [53].

Let K be a continuous, symmetric, positive-definite function (a 'kernel') $K : X \times X \to \mathbb{R}$ and set $K_x := K(\cdot, x)$. Define the Hilbert space $\mathcal{H}_K$ by first considering all finite linear combinations of

[7] Since $\sqrt{Y^T(K(X,X))^{-1}Y}$ is only a lower bound on $||f^\dagger||_K$, this modified estimate is no longer a rigorous bound. We also note that the effect of KF is to decrease both $\sqrt{Y^T(K(X,X))^{-1}Y}$ and $||f^\dagger||_K$.

functions K_x, that is, $\sum_{x_i \in X} a_i K_{x_i}$ with finitely many $a_i \in \mathbb{R}$ non-zero. An inner product $\langle \cdot, \cdot \rangle_K$ on this space is defined by $\langle K_{x_i}, K_{x_j} \rangle_H := K(x_i, x_j)$ and extends linearly. One takes the completion to obtain $\mathcal{H}_K$.

Alternatively, an equivalent definition of an RKHS is as a Hilbert space of real-valued functions on X for which the evaluation functional $\delta_x(f) := f(x)$ is continuous for all $x \in X$.

Finite-dimensional subspaces of $\mathcal{H}_K$ can also be naturally defined by taking a finite number of points $\mathbf{x} := \{x_1, \ldots, x_m\} \subset X$ and considering the linear span $\mathcal{H}_{K,\mathbf{x}} := \mathrm{span}\{K_x : x \in \mathbf{x}\}$. In practice, we will seek functions in these finite-dimensional subspaces as approximations for f^*.

RKHSs are characterized by the reproducing property

$$\langle K_x, f \rangle_K = f(x), \quad \forall f \in \mathcal{H}_K. \tag{B 3}$$

If we denote $\kappa := \sqrt{\sup_{x \in X} K(x, x)}$, then $\mathcal{H}_K \subset C(X)$ and it follows that

$$||f||_{L^\infty(X)} \le \kappa ||f||_K, \quad \forall f \in \mathcal{H}_K. \tag{B 4}$$

The RKHS $\mathcal{H}_K$ can also be defined by means of an integral operator. Let ρ be any (finite) strictly positive Borel measure on X (e.g. Lebesgue measure) and $L^2_\rho(X)$ be the Hilbert space of square integrable functions on X. Then define the linear operator $L_K : L^2_\rho(X) \to C(X)$ by $(L_K f)(x) = \int_X K(x, y) f(y) \mathrm{d}\rho(y)$. The RKHS will then correspond to the closure of the span of the eigenfunctions of L_K.

Theorem B.1. *Let K be a Mercer kernel such that $K \in C^{2s+\epsilon}(X \times X)$ for $0 < \epsilon < 2$ and suppose that $L_K^{-r} f^* \in \mathcal{L}^2_\rho(X)$ for some $0 < r \le 1$. Let $\lambda > 0$, then for every $0 < \delta < 1$, with probability $1 - \delta$, we have*

$$||f_{z,\lambda} - f^*||_\infty \le \frac{||w||_{\mathbf{R}^m} \sigma \kappa^2}{\lambda \sqrt{\delta}} + \frac{\kappa C h_{\mathbf{x}} \rho(X)}{\lambda} + \kappa \lambda^{r-(1/2)} ||L_K^{-r} f^*||_\infty,$$

where $\sigma^2 := \sup_{x \in X} \sigma^2_{x_i}$, $\kappa := \sqrt{\sup_{x \in X} K(x, x)}$ and C is a Lipschitz constant for $(f^* - f_\lambda)$, where the constant C depends on f^*, d, σ and the choice of RKHS $\mathcal{H}_{K^1}$.

The three terms in theorem B.1 correspond, respectively, to errors incurred by the *noise* (*sample error*, $\mathcal{E}_1$), the *finite set of data sites* (*integration error*, $\mathcal{E}_2$) and the *regularization parameter* λ (*regularization error*, $\mathcal{E}_3$).

References

1. Huntingford C, Jeffers ES, Bonsall MB, Christensen HM, Lees T, Yang H. 2019 Machine learning and artificial intelligence to aid climate change research and preparedness. *Environ. Res. Lett.* **14**, 124007. (doi:10.1088/1748-9326/ab4e55)
2. O'Gorman PA, Dwyer JG. 2018 Using machine learning to parameterize moist convection: potential for modeling of climate, climate change, and extreme events. *J. Adv. Model. Earth Syst.* **10**, 2548–2563. (doi:10.1029/2018MS001351)
3. Rolnick D *et al.* 2019 Tackling climate change with machine learning. (https://arxiv.org/abs/1906.05433)
4. Schmidt OT, Mengaldo G, Balsamo G, Wedi NP. 2019 Spectral empirical orthogonal function analysis of weather and climate data. *Mon. Weather Rev.* **147**, 2979–2995. (doi:10.1175/MWR-D-18-0337.1)
5. Prabhat M *et al.* 2017 ClimateNet: a machine learning dataset for climate science research. In *AGU fall meeting abstracts*, vol. 2017, p. IN13E-01. Washington, DC: American Geophysical Union.
6. Scher S. 2018 Toward data-driven weather and climate forecasting: approximating a simple general circulation model with deep learning. *Geophys. Res. Lett.* **45**, 12 616–12 622. (doi:10.1029/2018GL080704)
7. Toms BA, Kashinath K, Yang D. 2019 Deep learning for scientific inference from geophysical data: the Madden-Julian oscillation as a test case. (https://arxiv.org/abs/1902.04621)
8. Van der Baan M, Jutten C. 2000 Neural networks in geophysical applications. *Geophysics* **65**, 1032–1047. (doi:10.1190/1.1444797)

9. Owhadi H, Yoo GR. 2019 Kernel flows: from learning kernels from data into the abyss. *J. Comput. Phys.* **389**, 22–47. (doi:10.1016/j.jcp.2019.03.040)
10. Hamzi B, Owhadi H. 2021 Learning dynamical systems from data: a simple cross-validation perspective, Part I: parametric kernel flows. *Physica D* **421**, 132817. (doi:10.1016/j.physd.2020.132817)
11. Domingos P. 2020 Every model learned by gradient descent is approximately a kernel machine. (https://arxiv.org/abs/2012.00152)
12. Montavon G, Braun ML, Muller K-R. 2011 Kernel analysis of deep networks. *J. Mach. Learn. Res.* **12**, 2563–2581.
13. Owhadi H. 2020 Do ideas have shape? Plato's theory of forms as the continuous limit of artificial neural networks. (https://arxiv.org/abs/2008.03920)
14. Lundberg SM, Lee S-I. 2017 A unified approach to interpreting model predictions. In *Proc. of the 31st Int. Conf. on Neural Information Processing Systems, Long Beach, CA, 4–9 December 2017*, pp. 4768–4777. Red Hook, NY: Curran Associates.
15. Bach S, Binder A, Montavon G, Klauschen F, Müller K-R, Samek W. 2015 On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **10**, e0130140. (doi:10.1371/journal.pone.0130140)
16. Dell M, Jones BF, Olken BA. 2012 Temperature shocks and economic growth: evidence from the last half century. *Am. Econ. J.: Macroecon.* **4**, 66–95. (doi:10.1257/mac.4.3.66)
17. Molteni F, Buizza R, Palmer TN, Petroliagis T. 1996 The ECMWF ensemble prediction system: methodology and validation. *Q. J. R. Meteorol. Soc.* **122**, 73–119. (doi:10.1002/qj.49712252905)
18. Saha S *et al.* 2010 The NCEP climate forecast system reanalysis. *Bull. Am. Meteorol. Soc.* **91**, 1015–1058. (doi:10.1175/2010BAMS3001.1)
19. Lakshminarayanan B, Pritzel A, Blundell C. 2016 Simple and scalable predictive uncertainty estimation using deep ensembles. (https://arxiv.org/abs/1612.01474)
20. Bolton T, Zanna L. 2019 Applications of deep learning to ocean data inference and subgrid parameterization. *J. Adv. Model. Earth Syst.* **11**, 376–399. (doi:10.1029/2018MS001472)
21. Brenowitz ND, Bretherton CS. 2018 Prognostic validation of a neural network unified physics parameterization. *Geophys. Res. Lett.* **45**, 6289–6298. (doi:10.1029/2018GL078510)
22. Chattopadhyay A, Subel A, Hassanzadeh P. 2020 Data-driven super-parameterization using deep learning: experimentation with multiscale Lorenz 96 systems and transfer learning. *J. Adv. Model. Earth Syst.* **12**, e2020MS002084. (doi:10.1029/2020MS002084).
23. Gentine P, Pritchard M, Rasp S, Reinaudi G, Yacalis G. 2018 Could machine learning break the convection parameterization deadlock? *Geophys. Res. Lett.* **45**, 5742–5751. (doi:10.1029/2018GL078202)
24. Rasp S, Pritchard MS, Gentine P. 2018 Deep learning to represent subgrid processes in climate models. *Proc. Natl Acad. Sci. USA* **115**, 9684–9689. (doi:10.1073/pnas.1810286115)
25. Zanna L, Bolton T. 2020 Data-driven equation discovery of ocean mesoscale closures. *Geophys. Res. Lett.* **47**, e2020GL088376. (doi:10.1029/2020GL088376)
26. Chattopadhyay A, Nabizadeh E, Hassanzadeh P. 2020 Analog forecasting of extreme-causing weather patterns using deep learning. *J. Adv. Model. Earth Syst.* **12**, e2019MS001958. (doi:10.1029/2019MS001958)
27. Liu Y *et al.* 2016 Application of deep convolutional neural networks for detecting extreme weather in climate datasets. (https://arxiv.org/abs/1605.01156)
28. Nooteboom PD, Feng QY, López C, Hernández-García E, Dijkstra HA. 2018 Using network theory and machine learning to predict El Niño. *Earth Syst. Dyn.* **9**, 969–983. (doi:10.5194/esd-9-969-2018)
29. Rasp S, Thuerey N. 2020 Purely data-driven medium-range weather forecasting achieves comparable skill to physical models at similar resolution. (https://arxiv.org/abs/2008.08626)
30. Rodrigues ER, Oliveira I, Cunha R, Netto M. 2018 Deepdownscale: a deep learning strategy for high-resolution weather forecast. In *Proc. 2018 IEEE 14th Int. Conf. on e-Science (e-Science), Amsterdam, The Netherlands, 29 October–1 November 2018*, pp. 415–422. New York, NY: IEEE.
31. Weyn JA, Durran DR, Caruana R. 2020 Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. *J. Adv. Model. Earth Syst.* **12**, e2020MS002109. (doi:10.1029/2020MS002109).
32. Dueben PD, Bauer P. 2018 Challenges and design choices for global weather and climate models based on machine learning. *Geosci. Model Dev.* **11**, 3999–4009. (doi:10.5194/gmd-11-3999-2018)

33. Maulik R, Egele R, Lusch B, Balaprakash P. 2020 Recurrent neural network architecture search for geophysical emulation. In *Proc. of the Int. Conf. for High Performance Computing, Networking, Storage and Analysis, SC '20, virtual meeting, 9–19 November 2020*. New York, NY: IEEE Press.
34. Shi X, Chen Z, Wang H, Yeung D-Y, Wong W-K, Woo W-C. 2015 Convolutional LSTM network: a machine learning approach for precipitation nowcasting. (https://arxiv.org/abs/1506.04214)
35. Skinner DJ, Maulik R. 2020 Meta-modeling strategy for data-driven forecasting. (https://arxiv.org/abs/2012.00678)
36. Karpatne A, Ebert-Uphoff I, Ravela S, Babaie HA, Kumar V. 2018 Machine learning for the geosciences: challenges and opportunities. *IEEE Trans. Knowl. Data Eng.* **31**, 1544–1554. (doi:10.1109/TKDE.2018.2861006)
37. Rasp S, Thuerey N. 2021 Data-driven medium-range weather prediction with a resnet pretrained on climate simulations: a new model for weatherbench. *J. Adv. Model. Earth Syst.* **13**, e2020MS002405. (doi:10.1029/2020MS002405)
38. Maulik R, Mohan A, Lusch B, Madireddy S, Balaprakash P, Livescu D. 2020 Time-series learning of latent-space dynamics for reduced-order model closure. *Physica D* **405**, 132368. (doi:10.1016/j.physd.2020.132368)
39. Portwood GD *et al.* 2019 Turbulence forecasting via neural ode. (https://arxiv.org/abs/1911.05180)
40. Callaham JL, Maeda K, Brunton SL. 2019 Robust flow reconstruction from limited measurements via sparse representation. *Phys. Rev. Fluids* **4**, 103907. (doi:10.1103/PhysRevFluids.4.103907)
41. Kutz JN, Fu X, Brunton SL. 2016 Multiresolution dynamic mode decomposition. *SIAM J. Appl. Dyn. Syst.* **15**, 713–735. (doi:10.1137/15M1023543)
42. Maulik R, Fukami K, Ramachandra N, Fukagata K, Taira K. 2020 Probabilistic neural networks for fluid flow surrogate modeling and data recovery. *Phys. Rev. Fluids* **5**, 104401. (doi:10.1103/PhysRevFluids.5.104401)
43. Taira K, Hemati MS, Brunton SL, Sun Y, Duraisamy K, Bagheri S, Dawson STM, Yeh C-A. 2020 Modal analysis of fluid flows: applications and outlook. *AIAA J.* **58**, 998–1022. (doi:10.2514/1.J058462)
44. Owhadi H, Scovel C. 2019 *Operator-adapted wavelets, fast solvers, and numerical homogenization: from a game theoretic approach to numerical approximation and algorithm design*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge, UK: Cambridge University Press.
45. Yoo GR, Owhadi H. 2020 Deep regularization and direct training of the inner layers of neural networks with kernel flows. (https://arxiv.org/abs/2002.08335)
46. Owhadi H, Scovel C, Yoo GR. 2021 *Kernel mode decomposition and the programming of kernels*. Cham, Switzerland: Springer.
47. Hurrell JW *et al.* 2013 The Community Earth System Model: a framework for collaborative research. *Bull. Am. Meteorol. Soc.* **94**, 1339–1360. (doi:10.1175/BAMS-D-12-00121.1)
48. Chassignet EP *et al.* 2009 US GODAE: global ocean prediction with the Hybrid Coordinate Ocean Model (HYCOM). *Oceanography* **22**, 64–75. (doi:10.5670/oceanog.2009.39)
49. Wu Z-M, Schaback R. 1993 Local error estimates for radial basis function interpolation of scattered data. *IMA J. Numer. Anal.* **13**, 13–27. (doi:10.1093/imanum/13.1.13)
50. Owhadi H. 2015 Bayesian numerical homogenization. *Multiscale Model. Simul.* **13**, 812–828. (doi:10.1137/140974596)
51. Bouvrie J, Hamzi B. 2017 Kernel methods for the approximation of nonlinear systems. *SIAM J. Control Optim.* **55**, 2460–2492. (doi:10.1137/14096815X)
52. Giesl P, Hamzi B, Rasmussen M, Webster KN. 2016 Approximation of Lyapunov functions from noisy data. (https://arxiv.org/abs/1601.01568)
53. Cucker F, Smale S. 2002 On the mathematical foundations of learning. *Bull. Am. Math. Soc.* **39**, 1–49. (doi:10.1090/S0273-0979-01-00923-5)