PROCEEDINGS A

rspa.royalsocietypublishing.org

Research

Cite this article: Vlachas PR, Byeon W, Wan ZY, Sapsis TP, Koumoutsakos P. 2018 Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proc. R. Soc. A* **474**: 20170844. http://dx.doi.org/10.1098/rspa.2017.0844

Received: 8 December 2017
Accepted: 25 April 2018

Subject Areas:
mechanical engineering, computational physics, artificial intelligence

Keywords:
data-driven forecasting, long short-term memory, Gaussian processes, T21 barotropic climate model, Lorenz 96

Author for correspondence:
Petros Koumoutsakos
e-mail: petros@ethz.ch

Electronic supplementary material is available online at https://dx.doi.org/10.6084/m9.figshare.c.4094249.

THE ROYAL SOCIETY PUBLISHING

Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks

Pantelis R. Vlachas[1], Wonmin Byeon[1], Zhong Y. Wan[2], Themistoklis P. Sapsis[2] and Petros Koumoutsakos[1]

[1]Chair of Computational Science, ETH Zurich, Clausiusstrasse 33, Zurich, CH-8092, Switzerland
[2]Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

TPS, 0000-0003-0302-0691; PRV, 0000-0002-3311-2100

We introduce a data-driven forecasting method for high-dimensional chaotic systems using long short-term memory (LSTM) recurrent neural networks. The proposed LSTM neural networks perform inference of high-dimensional dynamical systems in their reduced order space and are shown to be an effective set of nonlinear approximators of their attractor. We demonstrate the forecasting performance of the LSTM and compare it with Gaussian processes (GPs) in time series obtained from the Lorenz 96 system, the Kuramoto–Sivashinsky equation and a prototype climate model. The LSTM networks outperform the GPs in short-term forecasting accuracy in all applications considered. A hybrid architecture, extending the LSTM with a mean stochastic model (MSM–LSTM), is proposed to ensure convergence to the invariant measure. This novel hybrid method is fully data-driven and extends the forecasting capabilities of LSTM networks.

1. Introduction

Natural systems, ranging from climate and ocean circulation to organisms and cells, involve complex dynamics extending over multiple spatio-temporal scales. Centuries-old efforts to comprehend and forecast the dynamics of such systems have spurred developments in large-scale simulations, dimensionality reduction techniques and a multitude of forecasting methods. The goals of understanding and prediction have been

complementing each other but have been hindered by the high dimensionality and chaotic behaviour of these systems. In recent years, we observe a convergence of these approaches due to advances in computing power, algorithmic innovations and the ample availability of data. A major beneficiary of this convergence are data-driven dimensionality reduction methods [1–7], model identification procedures [8–15] and forecasting techniques [16–30] that aim to provide precise short-term predictions while capturing the long-term statistics of these systems. Successful forecasting methods address the highly nonlinear energy transfer mechanisms between modes not captured effectively by the dimensionality reduction methods.

The pioneering technique of analogue forecasting proposed in [31] inspired a widespread research in non-parametric prediction approaches. Two dynamical system states are called analogues if they resemble one another on the basis of a specific criterion. This class of methods uses a training set of historical observations of the system. The system evolution is predicted using the evolution of the closest analogue from the training set corrected by an error term. This approach has led to promising results in practice [32] but the selection of the resemblance criterion to pick the optimal analogue is far from straightforward. Moreover, the geometrical association between the current state and the training set is not exploited. More recently [33], analogue forecasting is performed using a weighted combination of data-points based on a localized kernel that quantifies the similarity of the new point and the weighted combination. This technique exploits the local geometry instead of selecting a single optimal analogue. Similar kernel-based methods, [34] use diffusion maps to globally parametrize a low-dimensional manifold capturing the slower time scales. Moreover, non-trivial interpolation schemes are investigated in order to encode the system dynamics in this reduced order space as well as map them to the full space (lifting). Although the geometrical structure of the data is taken into account, the solution of an eigen-system with a size proportional to the training data is required, rendering the approach computationally expensive. In addition, the inherent uncertainty due to sparse observations in certain regions of the attractor introduces prediction errors which cannot be modelled in a deterministic context. In [35], a method based on Gaussian process regression (GPR) [36] was proposed for prediction and uncertainty quantification in the reduced order space. The technique is based on a training set that sparsely samples the attractor. Stochastic predictions exploit the geometrical relationship between the current state and the training set, assuming a Gaussian prior over the modelled latent variables. A key advantage of GPR is that uncertainty bounds can be analytically derived from the hyper-parameters of the framework. Moreover, in [35] a mean stochastic model (MSM) is used for under-sampled regions of the attractor to ensure accurate modelling of the steady state in the long-term regime. However, the resulting inference and training have a quadratic cost in terms of the number of data samples $O(N^2)$.

Some of the earlier approaches to capture the evolution of time series in chaotic systems using recurrent neural networks (RNNs) were developed during the inception of the long short-term memory networks (LSTM) [37]. However, to the best of our knowledge, these methods have been used only on low-dimensional chaotic systems [38]. Similarly, other machine learning techniques based on multi-layer perceptrons (MLP) [39], echo state networks (ESNs) [40,41] or radial basis functions [42,43] have been successful, albeit only for low-order dynamical systems. Recent work in [44,45] demonstrated promising results of ESNs for high-dimensional chaotic systems.

In this paper, we propose LSTM-based methods that exploit information of the recent history of the reduced order state to predict the high-dimensional dynamics. Time-series data are used to train the model while no knowledge of the underlying system equations is required. Inspired by Taken's theorem [46] an embedding space is constructed using time-delayed versions of the reduced order variable. The proposed method tries to identify an approximate forecasting rule globally for the reduced order space. In contrast with GPR [35], the method has a deterministic output while its training cost scales linearly with the number of training samples and it exhibits an $\mathcal{O}(1)$ inference computational cost. Moreover, following [35], LSTM is combined with a MSM, to cope with attractor regions that are not captured in the training set. In attractor regions, under-represented in the training set, the MSM is used to guarantee convergence to the invariant measure and avoid an exponential growth of the prediction error. The effectiveness

of the proposed hybrid method in accurate short-term prediction and capturing the long-term behaviour is shown in the Lorenz 96 system and the Kuramoto–Sivashinsky (K-S) system. Finally, the method is also tested on predictions of a prototypical climate model.

The structure of the paper is as follows: In §2, we explain how the LSTM can be employed for modelling and prediction of a reference dynamical system and a blended LSTM–MSM technique is introduced. In §3, three other state-of-the-art methods, GPR, MSM and the hybrid GPR-MSM scheme are presented and two comparison metrics are defined. The proposed LSTM technique and its LSTM–MSM extension are benchmarked against GPR and GPR–MSM in three complex chaotic systems in §4. In §5, we discuss the computational complexity of training and inference in LSTM. Finally, §6 offers a summary and discusses future research directions.

2. Long short-term memory recurrent neural networks

The LSTM was introduced in order to regularize the training of RNNs [37]. RNNs contain loops that allow information to be passed between consecutive temporal steps (figure 1*a*) and can be expressed as

$$\mathbf{h}_t = \sigma_{\mathrm{h}}(\mathbf{W}_{\mathrm{hi}}\mathbf{i}_t + \mathbf{W}_{\mathrm{hh}}\mathbf{h}_{t-1} + b_{\mathrm{h}}) \tag{2.1}$$

and

$$\mathbf{o}_t = \sigma_{\mathrm{o}}(\mathbf{W}_{\mathrm{oh}}\mathbf{h}_t + b_{\mathrm{o}}) = f^{w}(\mathbf{i}_t, \mathbf{h}_{t-1}), \tag{2.2}$$

where $\mathbf{i}_t \in \mathbb{R}^{d_i}$, $\mathbf{o}_t \in \mathbb{R}^{d_{\mathrm{o}}}$ and $\mathbf{h}_t \in \mathbb{R}^{d_{\mathrm{h}}}$ are the input, the output and the hidden state of the RNN at time step t. The weight matrices are $\mathbf{W}_{\mathrm{hi}} \in \mathbb{R}^{d_{\mathrm{h}} \times d_i}$ (input-to-hidden), $\mathbf{W}_{\mathrm{hh}} \in \mathbb{R}^{d_{\mathrm{h}} \times d_{\mathrm{h}}}$ (hidden-to-hidden), $\mathbf{W}_{\mathrm{oh}} \in \mathbb{R}^{d_{\mathrm{o}} \times d_{\mathrm{h}}}$ (hidden-to-output), b_{h} and b_{o}. Moreover, σ_{h} and σ_{o} are the hidden and output activation functions, while $b_{\mathrm{h}} \in \mathbb{R}^{d_{\mathrm{h}}}$ and $b_{\mathrm{o}} \in \mathbb{R}^{d_{\mathrm{o}}}$ are the respective biases. Temporal dependencies are captured by the hidden-to-hidden weight matrix $\mathbf{W}_{\mathrm{hh}}$, which couples two consecutive hidden states together. A schematic of the RNN architecture is given in figure 1.

In many practical applications, RNNs suffer from the vanishing (or exploding) gradient problem and have failed to capture long-term dependencies [47,48]. Today the RNNs owe their renaissance largely to the LSTM, that copes effectively with the aforementioned problem using *gates*. The LSTM has been successfully applied in sequence modelling [49], speech recognition [50], hand-writing recognition [51] and language translation [52].

The equations of the LSTM are

$$\left.\begin{aligned} &g_t^{\mathrm{f}} = \sigma_f(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{i}_t] + b_f) \quad g_t^{\mathrm{i}} = \sigma_i(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{i}_t] + b_i), \\ &\tilde{C}_t = \tanh(\mathbf{W}_C[\mathbf{h}_{t-1}, \mathbf{i}_t] + b_C) \quad C_t = g_t^{\mathrm{f}} C_{t-1} + g_t^{\mathrm{i}} \tilde{C}_t \\ \text{and} \quad &g_t^{o} = \sigma_{\mathrm{h}}(\mathbf{W}_{\mathrm{h}}[\mathbf{h}_{t-1}, \mathbf{i}_t] + b_{\mathrm{h}}) \quad \mathbf{h}_t = g_t^{o} \tanh(C_t), \end{aligned}\right\} \tag{2.3}$$

where $g_t^{\mathrm{f}}, g_t^{\mathrm{i}}, g_t^{\mathrm{o}} \in \mathbb{R}^{d_{\mathrm{h}} \times (d_{\mathrm{h}} + d_i)}$ are the gate signals (forget, input and output gates), $\mathbf{i}_t \in \mathbb{R}^{d_i}$ is the input, $\mathbf{h}_t \in \mathbb{R}^{d_{\mathrm{h}}}$ is the hidden state, $\mathbf{C}_t \in \mathbb{R}^{d_{\mathrm{h}}}$ is the cell state, while $\mathbf{W}_f$, $\mathbf{W}_i$, $\mathbf{W}_C, \mathbf{W}_{\mathrm{h}} \in \mathbb{R}^{d_{\mathrm{h}} \times (d_{\mathrm{h}} + d_i)}$ are weight matrices and $b_f, b_i, b_C, b_{\mathrm{h}} \in \mathbb{R}^{d_{\mathrm{h}}}$ are biases. The activation functions σ_f, σ_i and σ_{h} are sigmoids. For a more detailed explanation on the LSTM architecture refer to [37]. In the following, we refer to the LSTM hidden and cell states ($\mathbf{h}_t$ and C_t) jointly as *LSTM states*. The dimension of these states is called the number of hidden units $h = d_{\mathrm{h}}$ and it controls the capability of the cell to encode history information. In practice, we want the output to have a specific dimension d_{o}. For this reason, a fully connected final layer without activation function $\mathbf{W}_{\mathrm{oh}} \in \mathbb{R}^{d_{\mathrm{o}} \times h}$ is added

$$\mathbf{o}_t = \mathbf{W}_{\mathrm{oh}}\mathbf{h}_t = f^{\mathrm{w}}(\mathbf{i}_t, \mathbf{h}_{t-1}, C_{t-1}), \tag{2.4}$$

where all parameters (weights and biases) are encoded in $w = \{\mathbf{W}_{\mathrm{f}}, \mathbf{W}_{\mathrm{i}}, \mathbf{W}_{\mathrm{C}}, \mathbf{W}_{\mathrm{h}}, b_{\mathrm{f}}, b_{\mathrm{i}}, b_{\mathrm{C}}, b_{\mathrm{h}}\}$ and f^{w} is the LSTM cell function that maps the previous *LSTM States* $\mathbf{h}_{t-1}, C_{t-1}$ and current input $\mathbf{i}_t$

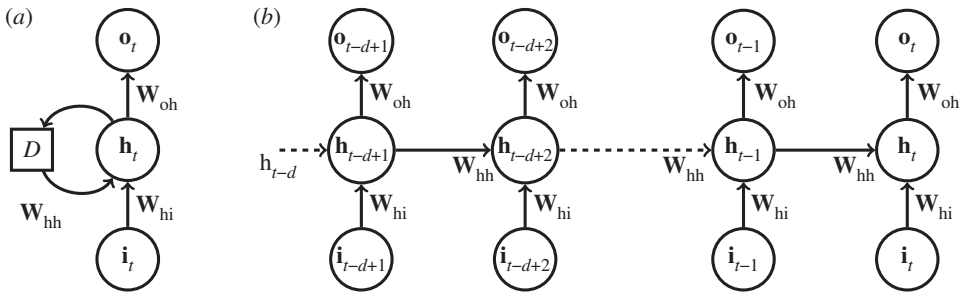

Figure 1. (*a*) A recurrent neural network cell, where D denotes a delay. The hidden cell state $\mathbf{h}_t$ depends on the input $\mathbf{i}_t$ and its previous value $\mathbf{h}_{t-1}$. The output $\mathbf{o}_t$ depends on the hidden state. The weight matrices are parameters of the cell. (*b*) A recurrent neural network unfolded in time (unfolding the delay). The same weights are used at each time step to compute the output $\mathbf{o}_t$ that depends on the current input $\mathbf{i}_t$ and short-term history (recursively) encoded in h_{t-1}.

to the output. By unfolding the LSTM d time-steps in the past and ignoring dependencies longer that d we get

$$\mathbf{o}_t = f^{\mathrm{w}}(z_t, \mathbf{h}_{t-1}, C_{t-1}) = \mathcal{F}^{w}(\underbrace{z_t, z_{t-1}, \ldots, z_{t-d+1}}_{z_{t:t-d+1}}, \cancelto{0}{\mathbf{h}_{t-d}}, \cancelto{0}{C_{t-d}}), \tag{2.5}$$

where $\mathcal{F}^{\mathrm{w}}$ represents the iterative application of f^{w} and computation of the *LSTM states* for d time steps. For a more detailed explanation of the formula for $\mathcal{F}^{\mathrm{w}}$, and a figure of the neural network architecture refer to the appendix.

In this work, we consider the reduced order problem where the state of a dynamical system is projected in the reduced order space. The system is considered to be autonomous, while $\dot{z}_t = \mathrm{d}z_t/\mathrm{d}t$ is the system state derivative at time step t. Following [38], the LSTM is trained using time series data from the system in the reduced order space $D = \{z_{1:T}, \dot{z}_{1:T}\}$ to predict the reduced state derivative $\dot{z}_t$ from a short history of the reduced order state $\{z_t, z_{t-1}, \ldots, z_{t-d+1}\}$ consisting of d past temporally consecutive states. In this work, we approximated the derivative from the original time series using first-order forward differences. The loss that has to be minimized is defined as

$$\mathcal{L}(D, w) = \frac{1}{T-d+1} \sum_{t=d}^{T} \| \underbrace{\mathcal{F}^{w}(z_{t:t-d+1})}_{\mathbf{o}_t} - \dot{z}_t \|^2. \tag{2.6}$$

The short-term history for the states before z_d is not available, that is why in total we have $T - d + 1$ training samples from a time series with T samples. During training the weights of the LSTM are optimized according to $w^\star = \underset{w}{\operatorname{argmin}}\, \mathcal{L}(D, w)$. The parameter d is denoted as truncation layer and time dependencies longer than d are not explicitly captured in the loss function.

Training of this model is performed using Back-propagation through time, truncated at layer d and mini-batch optimization with the Adam method [53] with an adaptive learning rate (initial learning rate $\eta = 0.0001$). The LSTM weights are initialized using the method of Xavier [54]. Training is stopped when convergence of the training error is detected or the maximum of 1000 epochs is reached. During training the loss of the model is evaluated on a separate validation dataset to avoid overfitting. The training procedure is explained in detail in the appendix.

An important issue is how to select the hidden state dimension h and how to initialize the *LSTM states* $\mathbf{h}_{t-d}, C_{t-d}$ at the truncation layer d. A small h reduces the expressive capabilities of the LSTM and deteriorates inference performance. On the other hand, a big h is more sensitive to overfitting and the computational cost of training rises. For this reason, h has to be tuned depending on the observed training behaviour. In this work, we performed a grid search and selected the optimal h for each application. For the truncation layer d, there are two alternatives, namely *stateless* and *stateful* LSTM. In *stateless* LSTM, the *LSTM states* at layer

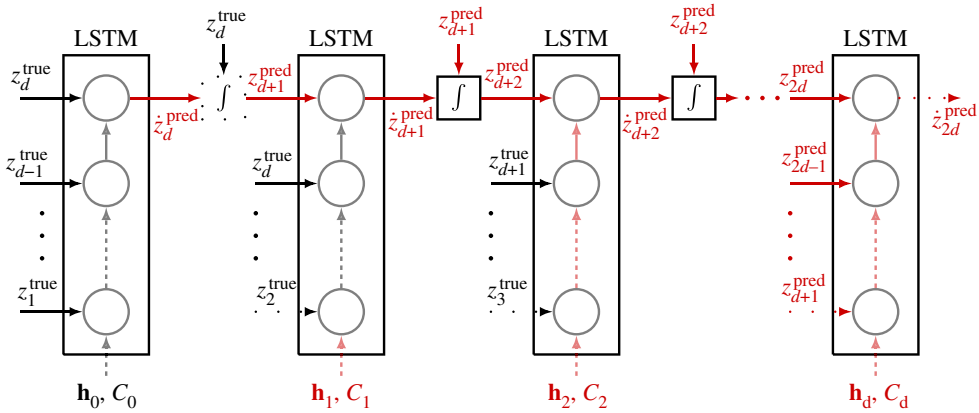

Figure 2. Iterative prediction using the trained LSTM model. A short-term history of the system, i.e. $z_1^{\text{true}}, \ldots, z_d^{\text{true}}$, is assumed to be known. Initial *LSTM states* are h_0, C_0. The trained LSTM is used predict the derivative $\dot{z}_d^{\text{pred}} = \mathcal{F}^w(z_{d:1}^{\text{true}}, \mathbf{h}_0, C_0)$. The state prediction z_{d+1}^{pred} is obtained by integrating this derivative. This value is used for the next prediction in an iterative fashion. After d time-steps only predicted values are fed in the input. In *stateless* LSTM, **h** and C are initialized to zero before every prediction. (Online version in colour.)

d are initialized to zero as in equation (2.5). As a consequence, the LSTM can only capture dependencies up to d previous time steps. In the second variant, the *stateful* LSTM, the state is always propagated for p time steps in the future and then reinitialized to zero, to help the LSTM capture longer dependencies. In this work, the systems considered exhibit chaotic behaviour and the dependencies are inherently short term, as the states in two time steps that differ significantly can be considered statistically independent. For this reason, the short temporal dependencies can be captured without propagating the hidden state for a long horizon. As a consequence, we consider only the *stateless* variant $p = 0$. We also applied *stateful* LSTM without any significant improvement so we omit the results for brevity. The trained LSTM model can be used to iteratively predict the system dynamics as illustrated in figure 2. This is a solely data-driven approach and no explicit information regarding the form of the underlying equations is required.

(a) Mean stochastic model and hybrid LSTM–MSM

The MSM is a powerful data-driven method used to quantify uncertainty and perform forecasts in turbulent systems with high intrinsic attractor dimensionality [35,55]. It is parametrized *a priori* to capture global statistical information of the attractor by design, while its computationally complexity is very low compared to LSTM or GPR. The concept behind MSM is to model each component of the state z^i independently with an Ornstein–Uhlenbeck (OU) process that captures the energy spectrum and the damping time scales of the statistical equilibrium. The process takes the following form:

$$\mathrm{d}z^i = c_i z^i \,\mathrm{d}z + \xi_i \,\mathrm{d}W_i, \tag{2.7}$$

where c_i, ξ_i are parameters fitted to the centred training data and W_i is a Wiener process. In the statistical steady state, the mean, energy and damping time scale of the process are given by

$$\mu_i = \mathbb{E}[z^i] = 0, \quad E_i = \mathbb{E}[z^i (z^i)^*] = -\frac{\xi^2}{2c_i} \quad \text{and} \quad T_i = -\frac{1}{c_i}, \tag{2.8}$$

where $(z^i)^*$ denotes the complex conjugate of z^i. To fit the model parameters c_i, ξ_i, we directly estimate the variance $\mathbb{E}[z^i(z^i)^*]$ from the time series training data and the decorrelation time using

$$T_i = \frac{1}{\mathbb{E}[z^i(z^i)^*]} \int_0^\infty \mathbb{E}[z^i(t)(z^i(t+\tau))^*] \,\mathrm{d}\tau. \tag{2.9}$$

After computing these two quantities, we replace in (2.8) and solve with respect to c_i and ξ_i. As the MSM is modelled *a priori* to mimic the global statistical behaviour of the attractor, forecasts made with MSM can never escape. This is not the case with LSTM and GPR, as prediction errors accumulate and iterative forecasts escape the attractor due to the chaotic dynamics, although short-term predictions are accurate. This problem has been addressed with respect to GPR in [35]. To cope effectively with this problem, we introduce a hybrid LSTM–MSM technique that prevents forecasts from diverging from the attractor.

The state-dependent decision rule for forecasting in LSTM–MSM is given by

$$\dot{z}_t = \begin{cases} (\dot{z}_t)_{\mathrm{LSTM}}, & \text{if } p^{\mathrm{train}}(z_t) = \prod p_i^{\mathrm{train}}(z_t^i) > \delta, \\ (\dot{z}_t)_{\mathrm{MSM}}, & \text{otherwise}, \end{cases} \tag{2.10}$$

where $p^{\mathrm{train}}(z_t)$ is an approximation of the probability density function of the training dataset and $\delta \approx 0.01$ a constant threshold tuned based on $p^{\mathrm{train}}(z_t)$. We approximate $p^{\mathrm{train}}(z_t)$ using a mixture of Gaussian kernels. This hybrid architecture exploits the advantages of LSTM and MSM. In case, there is a high probability that the state z_i lies close to the training dataset (interpolation) the LSTM having memorized the local dynamics is used to perform inference. This ensures accurate LSTM short-term predictions. On the other hand, close to the boundaries the attractor is only sparsely sampled $p^{\mathrm{train}}(z_i) < \delta$ and errors from LSTM predictions would lead to divergence. In this case, MSM guarantees that forecasting trajectories remain close to the attractor, and that we converge to the statistical invariant measure in the long term.

3. Benchmark and performance measures

The performance of the proposed LSTM-based prediction mechanism is benchmarked against the following state-of-the-art methods:

— MSM
— GPR
— mixed model (GPR–MSM)

To guarantee that the prediction performance is independent of the initial condition selected, for all applications and all performance measures considered the average value of each measure for a number of different initial conditions sampled independently and uniformly from the attractor is reported. The ground truth trajectory is obtained by integrating the discretized reference equation starting from each initial condition, and projecting the states to the reduced order space. The reference equation and the projection method are of course application dependent.

From each initial condition, we generate an empirical Gaussian ensemble of dimension N_{en} around the initial condition with a small variance σ_{en}. This noise represents the uncertainty in the knowledge of the initial system state. We forecast the evolution of the ensemble by iteratively predicting the derivatives and integrating (deterministically for each ensemble member for the LSTM, stochastically for GPR) and we keep track of the mean. We select an ensemble size $N_{\mathrm{en}} = 50$, which is the usual choice in environmental science, e.g. weather prediction and short-term climate prediction [56].

The ground truth trajectory at each time instant z is then compared with the predicted ensemble mean $\tilde{z}$. As a comparison measure we use the root mean square error (RMSE) defined as $\mathrm{RMSE}(z_k) = \sqrt{1/V \sum_{i=1}^{V} (z_k^{\mathrm{i}} - \tilde{z}_k^i)^2}$, where index k denotes the kth component of the reduced order state z, i is the initial condition, and V is the total number of initial conditions. The RMSE is computed at each time instant for each component k of the reduced order state, resulting in error curves that describe the evolution of error with time. Moreover, we use the standard deviation σ of the attractor samples in each dimension as a relative comparison measure. Assuming that the attractor consists of samples $\{z_1, z_2, \ldots, z_N\}$, with $z_j \in \mathbb{R}^{d_i}$, the attractor standard deviation in

dimension $i \in \{1, \ldots, d_i\}$ is defined as $\sigma_i = \sqrt{\mathbb{E}[(z^i - \bar{z}^i)^2]}$, where $\bar{z}^i$ is the mean of the samples in this dimension. If the prediction error is bigger than this standard deviation, then a trivial mean predictor performs better.

Moreover, we use the mean anomaly correlation coefficient (ACC) [57] over V initial conditions to quantify the pattern correlation of the predicted trajectories with the ground-truth. The ACC is defined as

$$\mathrm{ACC} = \frac{1}{V} \sum_{i=1}^{V} \frac{\sum_{k=1}^{r_{\mathrm{dim}}} w_k (z_k^{\mathrm{i}} - \bar{z}_k)(\tilde{z}_k^i - \bar{z}_k)}{\sqrt{\sum_{k=1}^{r_{\mathrm{dim}}} w_k (z_k^{\mathrm{i}} - \bar{z}_k)^2 \sum_{k=1}^{r_{\mathrm{dim}}} w_k (\tilde{z}_k^i - \bar{z}_k)^2}}, \tag{3.1}$$

where k refers to the mode number, i refers to the initial condition, w_k are mode weights selected according to the energies of the modes after dimensionality reduction and $\bar{z}_k$ is the time average of the respective mode, considered as reference. This score ranges from -1.0 to 1.0. If the forecast is perfect, the score equals to 1.0. The ACC coefficient is a widely used forecasting accuracy score in the meteorological community [58].

4. Applications

In this section, the effectiveness of the proposed method is demonstrated with respect to three chaotic dynamical systems, exhibiting different levels of chaos, from weakly chaotic to fully turbulent, i.e. the Lorenz 96 system, the K-S equation and a prototypical barotropic climate model.

(a) The Lorenz 96 system

In [59], a model of the large-scale behaviour of the mid-latitude atmosphere is introduced. This model describes the time evolution of the components X_j for $j \in \{0, 1, \ldots, J-1\}$ of a spatially discretized (over a single latitude circle) atmospheric variable. In the following, we refer to this model as the Lorenz 96. The Lorenz 96 is usually used ([35,58] and references therein) as a toy problem to benchmark methods for weather prediction.

The system of differential equations that governs the Lorenz 96 is defined as

$$\frac{\mathrm{d}X_j}{\mathrm{d}t} = (X_{j+1} - X_{j-2})X_{j-1} - X_j + F, \tag{4.1}$$

for $j \in \{0, 1, \ldots, J-1\}$, where by definition $X_{-1} = X_J, X_{-2} = X_{J-1}$. In our analysis $J = 40$. The right-hand side of (4.1) consists of a nonlinear adjective term $(X_{j+1} - X_{j-2})X_{j-1} - X_j$, a linear advection (dissipative) term $-X_j$ and a positive external forcing term F. The discrete energy of the system remains constant throughout time and the Lorenz 96 states X_j remain bounded. By increasing the external forcing parameter F the behaviour that the system exhibits changes from periodic $F < 1$ to weakly chaotic ($F = 4$) to end up in fully turbulent regimes ($F = 16$). These regimes can be observed in figure 3.

Following [35,56], we apply a shifting and scaling to standardize the Lorenz 96 states X_j. The discrete or Dirichlet energy is given by $E = \frac{1}{2}\sum_{j=1}^{J} X_j^2$. In order for the scaled Lorenz 96 states to have zero mean and unit energy we transform them using

$$\tilde{X}_j = \frac{X_j - \bar{X}}{\sqrt{E_p}}, \quad \mathrm{d}\tilde{t} = \sqrt{E_p}\,\mathrm{d}t \quad \text{and} \quad E_p = \frac{1}{2T}\sum_{j=0}^{J-1}\int_{T_0}^{T_0+T} (X_j - \bar{X})^2\,\mathrm{d}t, \tag{4.2}$$

where E_p is the average energy fluctuation. In this way, the scaled energy is $\tilde{E} = \frac{1}{2}\sum_{j=0}^{J-1}\tilde{X}_j^2 = 1$ and the scaled variables have zero mean $\bar{\tilde{X}} = (1/J)\sum_{j=0}^{J-1}\tilde{X}_j = 0$, with $\bar{X}$ the mean state. The scaled Lorenz 96 states $\tilde{X}_j$ obey the following differential equation:

$$\frac{\mathrm{d}\tilde{X}_j}{\mathrm{d}\tilde{t}} = \frac{F - \bar{X}}{E_p} + \frac{(\tilde{X}_{j+1} - \tilde{X}_{j-2})\bar{X} - \tilde{X}_j}{\sqrt{E_p}} + (\tilde{X}_{j+1} - \tilde{X}_{j-2})\tilde{X}_{j-1}. \tag{4.3}$$

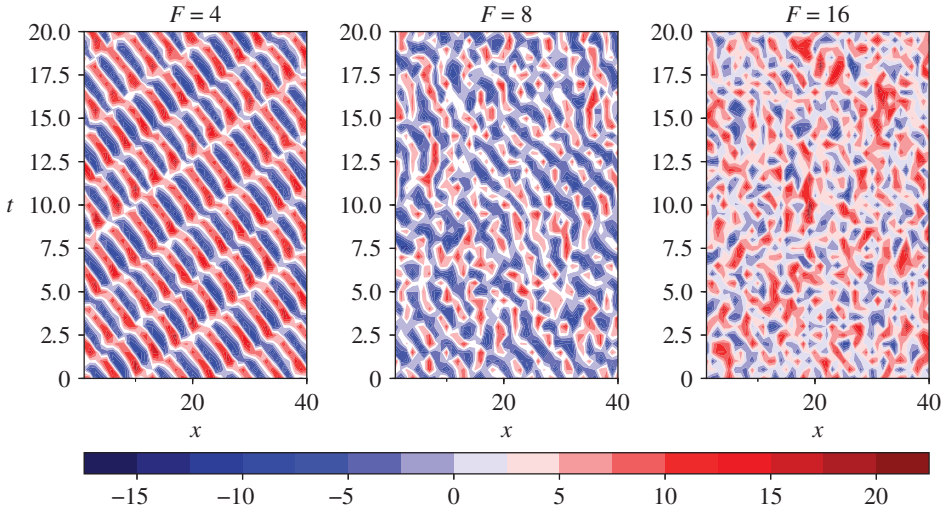

Figure 3. Lorenz 96 contour plots for different forcing regimes F. Chaoticity rises with bigger values of F.

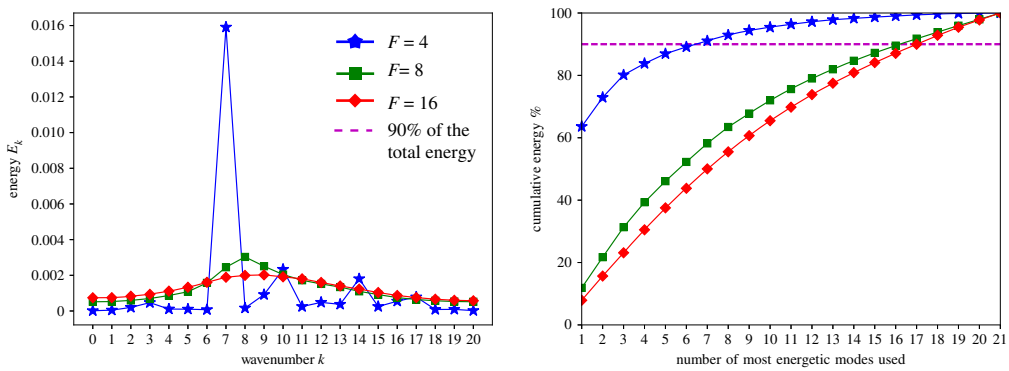

Figure 4. Energy spectrum E_k and cumulative energy with respect to the number of most energetic modes used for different forcing regimes of Lorenz 96 system. As the forcing increases, more chaoticity is introduced to the system. (Online version in colour.)

(i) Dimensionality reduction: discrete Fourier transform

Firstly, the discrete Fourier transform (DFT) is applied to the energy standardized Lorenz 96 states $\tilde{X}_j$. The Fourier coefficients $\hat{X}_k \in \mathbb{C}$ and the inverse DFT to recover the Lorenz 96 states are given by

$$\hat{X}_k = \frac{1}{J}\sum_{j=0}^{J-1} \tilde{X}_j \, \mathrm{e}^{-2\pi \mathrm{i} kj/J} \quad \text{and} \quad \tilde{X}_j = \sum_{k=0}^{J-1} \hat{X}_k \, \mathrm{e}^{2\pi \mathrm{i} kj/J}. \tag{4.4}$$

After applying the DFT to the Lorenz 96 states we end up with a symmetric energy spectrum that can be uniquely characterized by $J/2+1$ (J is considered to be an even number) coefficients $\hat{X}_k$ for $k \in K = \{0, 1, \ldots, J/2\}$. In our case $J = 40$, thus we end up with $|K| = 21$ complex coefficients $\hat{X}_k \in \mathbb{C}$. These coefficients are referred to as the Fourier modes or simply modes. The Fourier energy of each mode is defined as $E_k = \mathrm{Var}(\hat{X}_k) = \mathbb{E}[(\hat{X}_k(\tilde{t}) - \bar{\hat{X}}_k)(\hat{X}_k(\tilde{t}) - \bar{\hat{X}}_k)^*]$.

The energy spectrum of the Lorenz 96 system is plotted in figure 4 for different values of the forcing term F. We take into account only the $r_{\mathrm{dim}} = 6$ modes corresponding to the highest energies and the rest of the modes are truncated. For the different forcing regimes $F = 4, 8, 16$,

the six most energetic modes correspond to approximately 89%, 52% and 43.8% of the total energy, respectively. The space where the reduced variables live in is referred to as the reduced order phase space and the most energetic modes are notated as $\hat{X}_k^r$ for $k \in \{1, \ldots, r_{\text{dim}}\}$. As shown in [60], the most energetic modes are not necessarily the ones that capture better the dynamics of the model. Including more modes, or designing a criterion to identify the most important modes in the reduced order space may boost prediction accuracy. However, in this work, we are not interested in an optimal reduced space representation, but rather in the effectiveness of a prediction model given this space. The truncated modes are ignored for now. Nevertheless, their effect can be modelled stochastically as in [35].

As each Fourier mode $\hat{X}_k^r$ is a complex number, it consists of a real part and an imaginary part. By stacking these real and imaginary parts of the r_{dim} truncated modes we end up with the $2\,r_{\text{dim}}$ dimensional reduced model state

$$\mathbf{X} \equiv [\text{Re}(\hat{X}_1^r), \ldots, \text{Re}(\hat{X}_{r_{\text{dim}}}^r), \text{Im}(\hat{X}_1^r), \ldots, \text{Im}(\hat{X}_{r_{\text{dim}}}^r)]^{\text{T}}. \tag{4.5}$$

Assuming that X_j^t for $j \in \{0, 1, \ldots, J-1\}$ are the Lorenz 96 states at time instant t, the mapping X_j^t, $\forall j \to \mathbf{X}$ is unique and the reduced model state of the Lorenz 96 has a specific vector value.

(ii) Training and prediction in Lorenz 96

The reduced Lorenz 96 system states $\mathbf{X}_t$ are considered as the true reference states z_t. The LSTM is trained to forecast the derivative of the reduced order state $\dot{z}_t$ as elaborated in §2. We use a *stateless LSTM* with $h = 20$ hidden units and the back-propagation truncation horizon set to $d = 10$.

To obtain training data for the LSTM, we integrate the Lorenz 96 system state, e.g. (4.1) starting from an initial condition X_j^0 for $j \in \{0, 1, \ldots, J-1\}$ using a Runge–Kutta fourth-order method with a time step $dt = 0.01$ up to $T = 51$. In this way, a time series $X_j^t, t \in \{0, 1, \ldots\}$ is constructed. We obtain the reduced order state time series $\mathbf{X}_t, t \in \{0, 1, \ldots\}$, using the DFT mapping $|X_t^j \,\forall j \to \mathbf{X}_t$. From this time series, we discard the first 10^4 initial time steps as initial transients, ending up with a time series with $N^{\text{train}} = 50\,000$ samples. A similar but independent process is repeated for the validation set.

(iii) Results

The trained LSTM models are used for prediction based on the iterative procedure explained in §2. In this section, we demonstrate the forecasting capabilities of LSTM and compare it with GPs. One hundred different initial conditions uniformly sampled from the attractor are simulated. For each initial condition, an ensemble with size $N_{\text{en}} = 50$ is considered by perturbing it with a normal noise with variance $\sigma_{\text{en}} = 0.0001$.

In figure 5*a–c*, we report the mean RMSE prediction error of the most energetic mode $\hat{X}_1^r \in \mathbb{C}$, scaled with $\sqrt{E_p}$ for the forcing regimes $F \in \{6, 8, 16\}$ for the first $N = 10$ time steps ($T = 0.1$). In the RMSE, the complex norm $\|v\|_2 = vv^*$ is taken into account. The 10% of the standard deviation of the attractor is also plotted for reference (10%σ). As F increases, the system becomes more chaotic and difficult to predict. As a consequence, the number of prediction steps that remain under the 10%σ threshold are decreased. The LSTM models extend this predictability horizon for all forcing regimes compared to GPR and MSM. However, when LSTM is combined with MSM the short-term prediction performance is compromised. Nevertheless, hybrid LSTM–MSM models outperform GPR methods in short-term prediction accuracy.

In figure 5*d–f*, the RMSE error for $T = 2$ is plotted. The standard deviation from the attractor σ is plotted for reference. We can observe that both GPR and LSTM diverge, while MSM and blended schemes remain close to the attractor in the long term as expected.

In figure 5*g–i*, the mean ACC over 1000 initial conditions is given. The predictability threshold of 0.6 is also plotted. After crossing this critical threshold, the methods do not predict better than a trivial mean predictor. For $F = 4$, GPR methods show inferior performance compared to LSTM approaches as analysed previously in the RMSE comparison. However, for $F = 8$ LSTM models

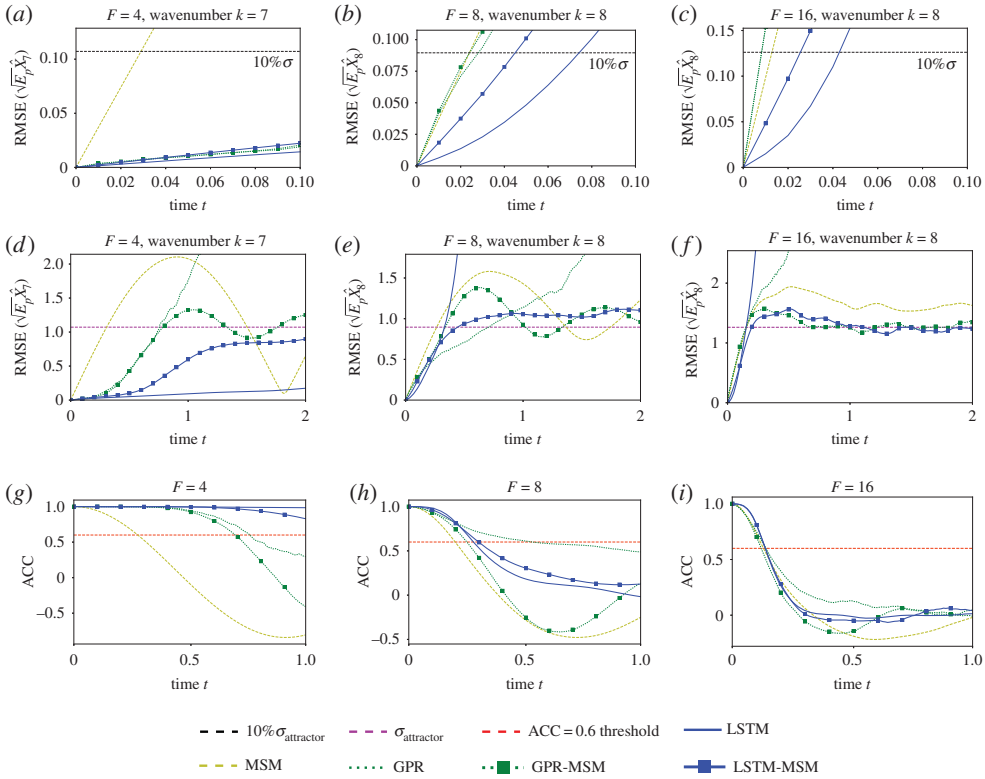

Figure 5. (*a–c*) Short-term RMSE evolution of the most energetic mode for forcing regimes $F = 4, 8, 16$, respectively, of the Lorenz 96 system. (*d–f*) Long-term RMSE evolution. (*g–i*) Evolution of the ACC coefficient (in all plots average over 1000 initial conditions is reported). (Online version in colour.)

do not predict better than the mean after $T \approx 0.35$, while GPR shows better performance. In turn, when blended with MSM the compromise in the performance for GPR–MSM is much bigger compared to LSTM–MSM. The LSTM–MSM scheme shows slightly superior performance than GPR–MSM during the entire relevant time period (ACC> 0.6). For the fully turbulent regime $F = 16$, LSTM shows comparable performance with both GPR and MSM and all methods converge as chaoticity rises, since the intrinsic dimensionality of the system attractor increases and the system becomes inherently unpredictable.

In figure 6, the evolution of the mean RMSE over 1000 initial conditions of the wavenumbers $k = 8, 9, 10, 11$ of the Lorenz 96 with forcing $F = 8$ is plotted. In contrast with GPR, the RMSE error of LSTM is much lower in the moderate and low energy wavenumbers $k = 9, 10, 11$ compared to the most energetic mode $k = 8$. This difference among modes is not observed in GPR. This can be attributed to the highly nonlinear energy transfer mechanisms between these lower energy modes as opposed to the Gaussian and locally linear energy transfers of the most energetic mode.

As illustrated before, the hybrid LSTM–MSM architecture effectively combines the accurate short-term prediction performance of LSTM with the long-term stability of MSM. The ratio of ensemble members modelled by LSTM in the hybrid scheme is plotted with respect to time in figure 7*a*. Starting from the initial ensemble of size 50, as the LSTM forecast might deviate from the attractor, the MSM is used to forecast in the hybrid scheme. As a consequence, the ratio of ensemble members modelled by LSTM decreases with time. In parallel with the GPR results presented in [35] and plotted in figure 7*b*, the slope of this ratio curve increases with F up to time $t \approx 1.5$. However, the LSTM ratio decreases slower compared to GPR.

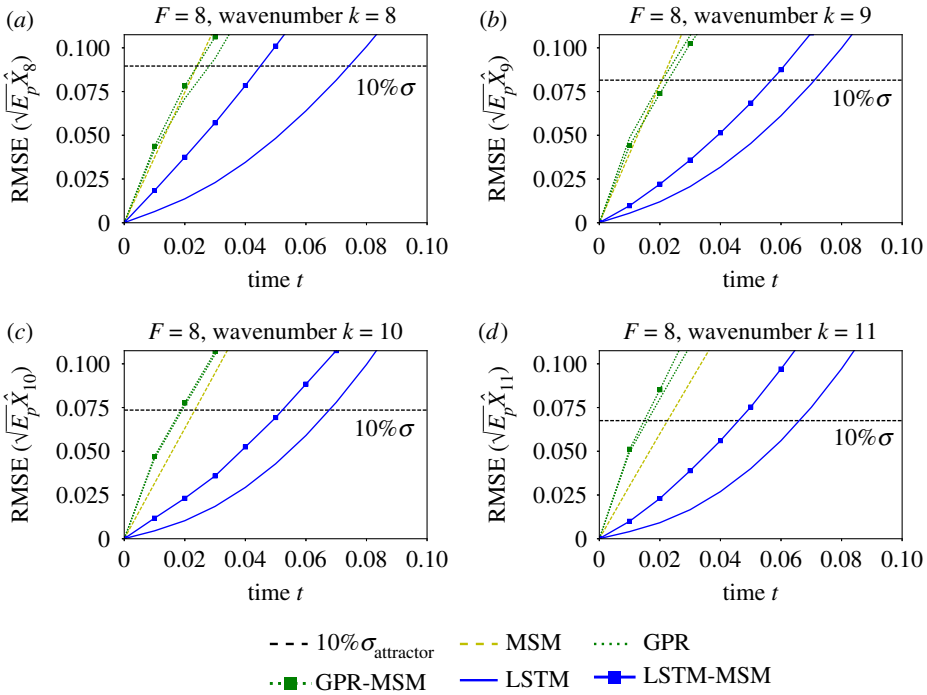

Figure 6. RMSE prediction error evolution of four energetic modes for the Lorenz 96 system with forcing $F = 8$. (*a*) Most energetic mode $k = 8$, (*b*) low-energy mode $k = 9$, (*c*) low-energy mode $k = 10$ and (*d*) low-energy mode $k = 11$ (in all plots average over 1000 initial conditions reported). (Online version in colour.)

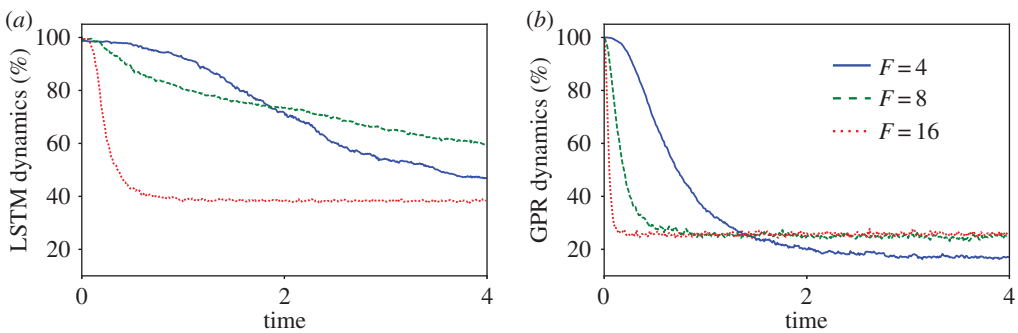

Figure 7. (*a*) Ratio of the ensemble members evaluated using the LSTM model over time for different Lorenz 96 forcing regimes in the hybrid LSTM–MSM method and (*b*) the same for GPR in the hybrid GPR–MSM method (average over 500 initial conditions). (Online version in colour.)

(b) Kuramoto–Sivashinsky equation

The K-S system is extensively used in many scientific fields to model a multitude of chaotic physical phenomena. It was first derived by Kuramoto [61,62] as a turbulence model of the phase gradient of a slowly varying amplitude in a reaction–diffusion type medium with negative viscosity coefficient. Later, Sivashinsky [63] studied the spontaneous instabilities of the plane front of a laminar flame ending up with the K-S equation, while in [64] the K-S equation is found to describe the surface behaviour of viscous liquid in a vertical flow.

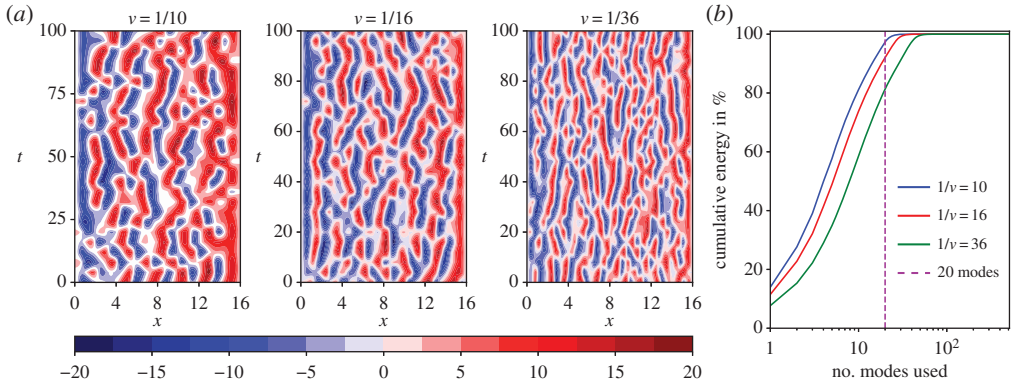

Figure 8. (*a*) Contour plots of the solution $u(x, t)$ of the Kuramoto–Sivashinsky system for different values of ν in steady state. Chaoticity rises with smaller values of ν. (*b*) Cumulative energy as a function of the number of the PCA modes for different values of ν. (Online version in colour.)

For our study, we restrict ourselves to the one-dimensional K-S equation with boundary and initial conditions given by

$$\left.\begin{aligned} &\frac{\partial u}{\partial t} = -\nu\frac{\partial^4 u}{\partial x^4} - \frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial x}, \\ &u(0,t) = u(L,t) = \left.\frac{\partial u}{\partial x}\right|_{x=0} = \left.\frac{\partial u}{\partial x}\right|_{x=L} = 0 \\ \text{and}\quad &u(x,0) = u_0(x), \end{aligned}\right\} \tag{4.6}$$

where $u(x, t)$ is the modelled quantity of interest depending on a spatial variable $x \in [0, L]$ and time $t \in [0, \infty)$. The negative viscosity is modelled by the parameter $\nu > 0$. We impose Dirichlet and second-type boundary conditions to guarantee ergodicity [65]. To spatially discretize (4.6) we use a grid size Δx with $D = L/\Delta x + 1$ the number of nodes. Further, we denote with $u_i = u(i\Delta x)$ the value of u at node $i \in \{0, \ldots, D-1\}$. Discretization using a second-order differences scheme yields

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = -\nu\frac{u_{i-2} - 4u_{i-1} + 6u_i - 4u_{i+1} + u_{i+2}}{\Delta x^4} - \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} - \frac{u_{i+1}^2 - u_{i-1}^2}{4\Delta x}. \tag{4.7}$$

Further, we impose $u_0 = u_{D-1} = 0$ and add ghost nodes $u_{-1} = u_1$, $u_D = u_{D-2}$ to account for the Dirichlet and second-order boundary conditions. In our analysis, the number of nodes is $D = 513$. The Kuramoto–Sivashinsky equation exhibits different levels of chaos depending on the bifurcation parameter $\tilde{L} = L/2\pi\sqrt{\nu}$ [66]. Higher values of $\tilde{L}$ lead to more chaotic systems in terms of higher Lyapunov exponents [35].

In our analysis, the spatial variable bound is held constant to $L = 16$ and chaoticity level is controlled through the negative viscosity ν, where a smaller value leads to a system with a higher level of chaos (figure 8*a*). In our study, we consider two values, namely $\nu = 1/10$ and $\nu = 1/16$ to benchmark the prediction skills of the proposed method. The discretized equation (4.7) is integrated with a time interval $dt = 0.02$ up to $T = 11\,000$. The data points up to $T = 1000$ are discarded as initial transients. Half of the remaining data ($N = 250\,000$ samples) are used for training and the other half for validation.

(i) Dimensionality reduction: singular value decomposition

The dimensionality of the problem is reduced using singular value decomposition (SVD). By subtracting the temporal mean $\overline{\mathbf{u}}$ and stacking the data, we end up with the data matrix

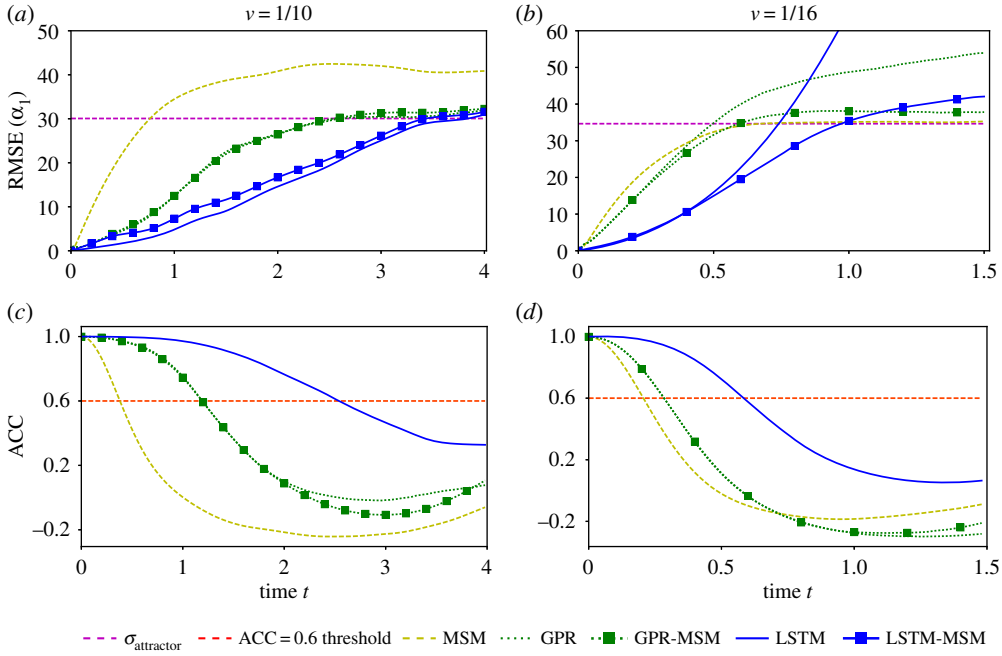

Figure 9. (*a*,*b*) RMSE evolution of the most energetic mode of the K-S equation with $1/\nu = 10$ and $1/\nu = 16$. (*c*), (*d*) ACC evolution of the most energetic mode of the K-S equation with $1/\nu = 10$ and $1/\nu = 16$ (in all plots, average value over 1000 initial conditions is reported). (Online version in colour.)

$\mathbf{U} \in \mathbb{R}^{N\times 513}$, where N is the number of data samples ($N = 500\,000$ in our case). Performing SVD on $\mathbf{U}$ leads to

$$\mathbf{U} = \mathbf{M}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}}, \quad \mathbf{M} \in \mathbb{R}^{N\times N}, \quad \boldsymbol{\Sigma} \in \mathbb{R}^{N\times 513} \quad \text{and} \quad \mathbf{V} \in \mathbb{R}^{513\times 513}, \tag{4.8}$$

with Σ diagonal, with descending diagonal elements. The right singular vectors corresponding to the r_{dim} largest singular values are the first columns of $\mathbf{V} = [\mathbf{V_r}, \mathbf{V_{-r}}]$. Stacking these singular vectors yields $\mathbf{V_r} \in \mathbb{R}^{513\times r_{\mathrm{dim}}}$. Assuming that $\mathbf{u}_t \in \mathbb{R}^{513}$ is a vector of the discretized values of $u(x,t)$ in time t, in order to get a reduced order representation $\mathbf{c} \equiv [c_1, \ldots, c_{r_{\mathrm{dim}}}]^{\mathrm{T}}$ corresponding to the r_{dim} components with the highest energies (singular values) we multiply

$$\mathbf{c} = \mathbf{V_r}^T\mathbf{u}, \quad \mathbf{c} \in \mathbb{R}^{r_{\mathrm{dim}}}. \tag{4.9}$$

The percentage of cumulative energy w.r.t. to the number of PCA modes considered is plotted in figure 8*b*. In our study, we pick $r_{\mathrm{dim}} = 20$ (out of 513) most energetic modes, as they explain approximately 90% of the total energy.

(ii) Results

We train *stateless* LSTM models with $h = 100$ and $d = 50$. For testing, starting from 1000 initial conditions uniformly sampled from the attractor, we generate a Gaussian ensemble of dimension $N_{\mathrm{en}} = 50$ centred around the initial condition in the original space with standard deviation of $\sigma = 0.1$. This ensemble is propagated using the LSTM prediction models, and GPR, MSM and GPR–MSM models trained as in [35]. The RMSE between the predicted ensemble mean and the ground-truth is plotted in figure 9*a*,*b* for different values of the parameter ν. All methods reach the invariant measure much faster for $1/\nu = 16$ compared to the less chaotic regime $1/\nu = 10$ (note the different integration times $T = 4$ for $1/\nu = 10$, while $T = 1.5$ for $1/\nu = 16$).

In both chaotic regimes $1/\nu = 10$ and $1/\nu = 16$, the reduced order LSTM outperforms all other methods in the short term before escaping the attractor. However, in the long term, LSTM

does not stabilize and will eventually diverge faster than GPR (figure 9*b*). Blending LSTM with MSM alleviates the problem and both accurate short-term predictions and long-term stability is attained. Moreover, the hybrid LSTM–MSM has better forecasting capabilities compared to GPR.

The need for blending LSTM with MSM in the KS equation is less imperative as the system is less chaotic than the Lorenz 96 and LSTM methods diverge much slower, while they sufficiently capture the complex nonlinear dynamics. As the intrinsic dimensionality of the attractor rises LSTM diverges faster.

The mean ACC (3.1) is plotted with respect to time in figure 9*c*,*d* for $\nu = 10$ and 16, respectively. The evolution of the ACC justifies the aforementioned analysis. The mean ACC of the trajectory predicted with LSTM remains above the predictability threshold of 0.6 for a highest time duration compared to other methods. This predictability horizon is approximately 2.5 for $\nu = \frac{1}{10}$ and 0.6 for $\nu = \frac{1}{16}$, since the chaoticity of the system rises and accurate predictions become more challenging. For a plot of the time evolution of the ratio of the ensemble members that are modelled with LSTM dynamics in the hybrid LSTM–MSM refer to the appendix.

(c) A barotropic climate model

In this section, we examine a standard barotropic climate model [67] originating from a realistic winter circulation. The model equations are given by

$$\frac{\partial \zeta}{\partial t} = -\mathcal{J}(\psi, \zeta + f + h) + k_1\zeta + k_2\delta^3\zeta + \zeta^*, \tag{4.10}$$

where ψ is the stream function, $\zeta = \delta\psi$ the relative vorticity, f the Coriolis parameter, ζ^* a constant vorticity forcing, while k_1 and k_2 are the Ekman damping and the scale-selective damping coefficient. $\mathcal{J}$ is the Jacobi operator given by

$$\mathcal{J}(a, b) = \left(\frac{\partial a}{\partial \lambda}\frac{\partial B}{\partial \mu} - \frac{\partial a}{\partial \mu}\frac{\partial B}{\partial \lambda}\right), \tag{4.11}$$

where μ and λ denote the sine of the geographical latitude and longitude, respectively. The equation of the barotropic model (4.10) is non-dimensionalized using the radius of the Earth as unit length and the inverse of the Earth angular velocity as time unit. The non-dimensional orography h is related to the real northern hemisphere orography h' by $h = 2\sin(\phi_0)A_0h'/H$, where phi_0 is a fixed amplitude of 45° N, A_0 is a factor expressing the surface wind strength blowing across the orography, and H a scale height [67]. The stream-function ψ is expanded into a spherical harmonics series and truncated at wavenumber 21, while modes with an even total wavenumber are excluded, avoiding currents across the equator and ending up with a hemispheric model with 231 degrees of freedom.

The training data are obtained by integrating the equation (4.10) for 10^5 days after an initial spin-up period of 1000 days, using a fourth-order Adams-Bashforth integration scheme with a 45-min time step in accordance with [35], with $k_1 = 15$ days, while k_2 is selected such that wavenumber 21 is damped at a time scale of 3 days. In this way, we end up with a time series ζ_t with 10^4 samples. The spherical surface is discretized into a $D = 64 \times 32$ mesh with equally spaces latitude and longitude. From the gathered data, 90% is used for training and 10% for validation. The mean and variance of the statistical steady state are shown in figure 10*a*,*b*.

The dimensionality of the barotropic climate model truncated at wavenumber 21 is 231. To reduce it, we identify empirical orthogonal functions (EOFs) ϕ_i, $i \in \{1, \ldots, 231\}$ that form an orthogonal basis of the reduced order space. The details of the method are described in the appendix. EOF analysis has been used to identify individual realistic climatic modes such as the Arctic oscillation (AO), the Pacific/North America (PNA) and the Tropical/Northern Hemisphere (TNH) pattern known as teleconnections [68,69]. Accurate prediction of these modes is of high practical importance as they feature realistic climate patterns. After projecting the state of the barotropic model to the EOFs, we take into account only the r_{dim} coefficients corresponding to the most energetic EOFs that form the reduced order state $\mathbf{y}^*$. In our study, the dimensionality of

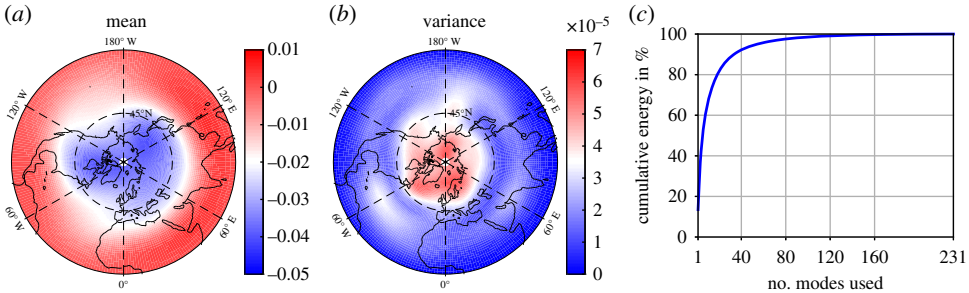

Figure 10. (*a*) Mean of the Barotropic model at statistical steady state. (*b*) Variance of the Barotropic model at statistical steady state. (*c*) Percentage of energy explained with respect to the modelled modes. (Online version in colour.)

the reduced space is $r_{\mathrm{dim}} = 30$, as $\boldsymbol{\phi}_{30}$ contains only 3.65% of the energy of $\boldsymbol{\phi}_1$, while the 30 most energetic modes contain approximately 82% of the total energy, as depicted in figure 10*c*.

(i) Training and prediction

The reduced order state that we want to predict using the LSTM are the 30 components of $\mathbf{y}$. A *stateless* LSTM with $h = 140$ hidden units is considered, while the truncated back-propagation horizon is set to $d = 10$. The prototypical system is less chaotic than the K-S equation and the Lorenz 96, which enables us to use more hidden units. The reason is that as chaoticity is decreased trajectories sampled from the attractor as training and validation dataset become more interconnected and the task is inherently easier and less prone to overfitting. In the extreme case of a periodic system, the information would be identical. Five hundred points are randomly and uniformly picked from the attractor as initial conditions for testing. A Gaussian ensemble with a small variance ($\sigma_{\mathrm{en}} = 0.001$) along each dimension is formed and marched using the reduced-order GPR, MSM, mixed GPR–MSM and LSTM methods.

(ii) Results

The RMSE error of the four most energetic reduced order space variables $\mathbf{y}_i$ for $i \in \{1, \ldots, 4\}$ is plotted in figure 11. The LSTM takes 400–500 h to reach the attractor, while GPR based methods generally take 300–400 h. By contrast, the MSM reaches the attractor already after 1 h. This implies that the LSTM can better capture the nonlinear dynamics compared to GPR. Note that the barotropic model is much less chaotic than the Lorenz 96 system with $F = 16$, where all methods show comparable prediction performance. Blended LSTM models with MSM are omitted here, as LSTM models only reach the attractor standard deviation towards the end of the simulated time and MSM–LSTM shows identical performance.

5. A comment on computational cost of prediction

The computational cost of making a single prediction can be quantified by the number of operations (multiplications and additions) needed. In GPR-based approaches the computational cost is of order $O(N^2)$, where N is the number of samples used in training. For GPR methods illustrated in the previous section $N \approx 2500$. The GPR models the global dynamics by uniformly sampling the attractor and 'carries' this training dataset at each time instant to identify the geometric relation between the input and the training dataset (modelled with a covariance matrix metric) and make (exact or approximate) probabilistic inference on the output.

By contrast, LSTM adjusts its parameters to reproduce the local dynamics. As a consequence, the inference computational complexity does not depend on the number of samples used for training. The inference complexity is roughly $O(d_i \cdot d \cdot h + d \cdot h^2)$, where d_i is the dimension of

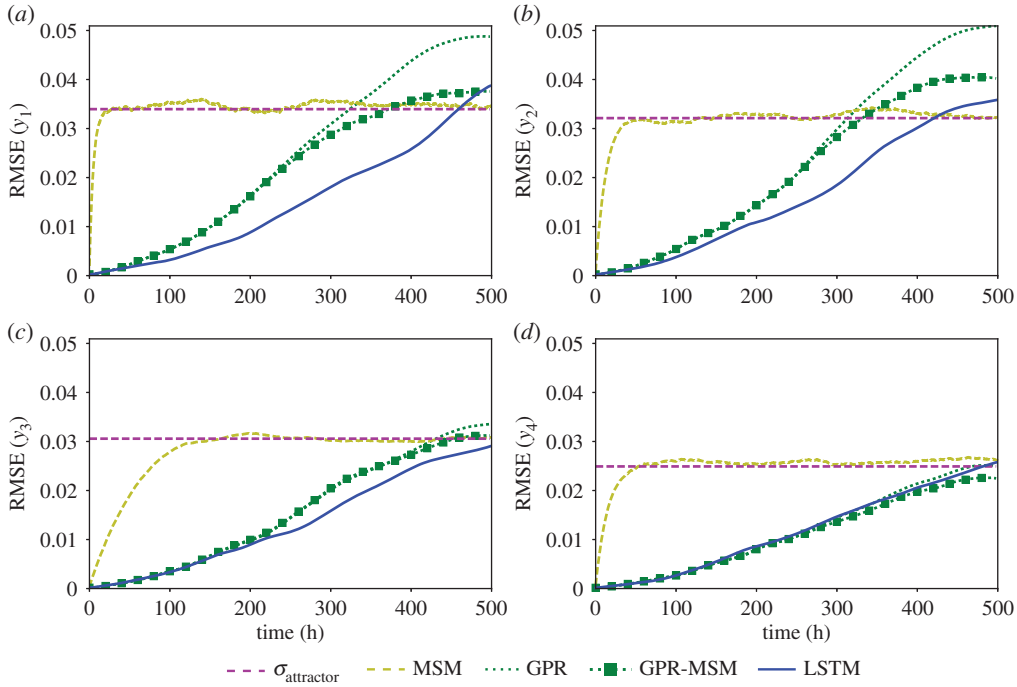

Figure 11. RMSE evolution of the four most energetic EOFs for the Barotropic climate model, average over 500 initial conditions reported. (*a*) Most energetic EOF, (*b*) second most energetic EOF, (*c*) third most energetic EOF and (*d*) fourth most energetic EOF. (Online version in colour.)

each input, d is the number of inputs and h is the number of hidden units. This complexity is significantly smaller than GPR, which can be translated to faster prediction. However, it is logical that the LSTM is more prone to diverge from the attractor, as there is no guarantee that the infrequent training samples near the attractor limits where memorized. This remark explains the faster divergence of LSTM in the more turbulent regimes considered in §4.

6. Conclusion

We propose a data-driven method, based on LSTM networks, for modelling and prediction in the reduced space of chaotic dynamical systems. The LSTM uses the short-term history of the reduced order variable to predict the state derivative and uses it for one-step prediction. The network is trained on time-series data and it requires no prior knowledge of the underlying governing equations. Using the trained network, long-term predictions are made by iteratively predicting one step forward.

The features of the proposed technique are showcased through comparisons with GPR and MSM on benchmarked cases. Three applications are considered, the Lorenz 96 system, the K-S equation and a barotropic climate model. The chaoticity of these systems ranges from weakly chaotic to fully turbulent, ensuring a complete simulation study. Comparison measures include the RMSE and ACC between the predicted trajectories and trajectories of the real dynamics.

In all cases, the proposed approach performs better, in short-term predictions, as the LSTM is more efficient in capturing the local dynamics and complex interactions between the modes. However, the prediction error accumulates as we iteratively perform predictions and similar to GPR does not converge to the invariant measure. Furthermore, in the cases of increased chaoticity the LSTM diverges faster than GPR. This may be attributed to the absence of certain attractor regions in the training data, insufficient training and propagation of the exponentially increasing

prediction error. To mitigate this effect, LSTM is also combined with MSM, following ideas presented in [35], in order to guarantee convergence to the invariant measure. Blending LSTM or GPR with MSM leads to a deterioration in the short-term prediction performance but the steady-state statistical behaviour is captured. The hybrid LSTM–MSM exhibits a slightly superior performance than GPR–MSM in all systems considered in this study.

In the K-S equation, LSTM can capture better the local dynamics compared to Lorenz 96 due to the lower intrinsic attractor dimensionality. LSTM is more accurate than GPR in the short term, but especially in the chaotic regime $1/\nu = 16$ forecasts of LSTM fly away from the attractor faster. LSTM–MSM counters this effect and long-term forecasts converge to the invariant measure at the expense of a compromise in the short-term forecasting accuracy. The higher short-term forecasting accuracy of LSTM can be attributed to the fact that it is a nonlinear approximator and can also capture correlations between modes in the reduced space. By contrast, GPR is a locally linear approximator modelling each mode independently in the output, assuming Gaussian correlations between modes in the input. LSTM and GPR show comparable forecasting accuracy in the barotropic model, as the intrinsic dimensionality is significantly smaller than K-S and Lorenz 96 and both methods can effectively capture the dynamics.

Future directions include modelling the lower energy modes and interpolation errors using a stochastic component in the LSTM to improve the forecasting accuracy. Another possible research direction is to model the attractor in the reduced space using a mixture of LSTM models, one model for each region. The LSTM proposed in this work models the attractor globally. However, different attractor regions may exhibit very different dynamic behaviours, which cannot be simultaneously modelled using only one network. Moreover, these local models can be combined with a closure scheme compensating for truncation and modelling errors. This local modelling approach may further improve prediction performance.

Data accessibility. The code and data used in this work are available at the link: https://polybox.ethz.ch/index.php/s/keH7PftvLmbkYU1. The password is `rspa_paper`. The `TensorFlow` library and `python3` were used for the implementation of LSTM architectures while `Matlab` was used for Gaussian Processes. These packages need to be installed in order to run the codes.

Authors' contributions. P.R.V. conceived the idea of the blended LSTM–MSM scheme, implemented the neural network architectures and the simulations, interpreted the computational results, and wrote the manuscript. W.B. supervised the work and contributed to the implementation of the LSTM. Z.Y.W. implemented the GPR and made contributions to the manuscript. P.K. had the original idea of the LSTM scheme and contributed to the manuscript. P.K. and T.P.S. contributed to the interpretation of the results and offered consultation. All the authors gave their final approval for publication.

Competing interests. We declare we have no competing interests.

Funding. T.P.S. and Z.Y.W. have been supported by an Air Force Office of Scientific Research grant no. FA9550-16-1-0231, an Office of Naval Research grant N00014-15-1-2381 and an Army Research Office grant no. 66710-EG-YIP. P.K. and P.R.V. gratefully acknowledge support from the European Research Council (ERC) Advanced Investigator Award (no. 341117).

Acknowledgements. We thank the two anonymous reviewers whose insightful comments helped us to enhance the manuscript.

References

1. Rowley CW. 2005 Model reduction for fluids, using balanced proper orthogonal decomposition. *Int. J. Bifurcation Chaos* **15**, 997–1013. (doi:10.1142/S0218127405012429)
2. Williams MO, Kevrekidis IG, Rowley CW. 2015 A data-driven approximation of the Koopman operator: extending dynamic mode decomposition. *J. Nonlinear Sci.* **25**, 1307–1346. (doi:10.1007/s00332-015-9258-5)
3. Tu JH, Rowley CW, Luchtenburg DM, Brunton SL, Kutz JN. 2014 On dynamic mode decomposition: theory and applications. *J. Comput. Dyn.* **1**, 391–421. (doi:10.3934/jcd.2014.1.391)
4. Kutz JN, Fu X, Brunton SL. 2016 Multiresolution dynamic mode decomposition. *SIAM. J. Appl. Dyn. Syst.* **15**, 713–735. (doi:10.1137/15M1023543)

5. Arbabi H, Mezic I. 2017 Ergodic theory, dynamic mode decomposition and computation of spectral properties of the Koopman operator. *SIAM. J. Appl. Dyn. Syst.* **16**, 2096–2126. (doi:10.1137/17M1125236)
6. Kerschen G, Golinval JC, Vakakis AF, Bergman LA. 2005 The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. *Nonlinear. Dyn.* **41**, 147–169. (doi:10.1007/s11071-005-2803-2)
7. Sapsis TP, Majda AJ. 2013 Statistically accurate low-order models for uncertainty quantification in turbulent dynamical systems. *Proc. Natl Acad. Sci. USA* **110**, 13 705–13 710. (doi:10.1073/pnas.1313065110)
8. Krischer K, Rico-Martínez R, Kevrekidis IG, Rotermund HH, Ertl G, Hudson JL. 1992 Model identification of a spatiotemporally varying catalytic reaction. *AIChE J.* **39**, 89–98. (doi:10.1002/aic.690390110)
9. Milano M, Koumoutsakos P. 2002 Neural network modeling for near wall turbulent flow. *J. Comput. Phys.* **182**, 1–26. (doi:10.1006/jcph.2002.7146)
10. Brunton SL, Proctor JL, Kutz JN. 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **113**, 3932–3937. (doi:10.1073/pnas.1517384113)
11. Duriez T, Brunton SL, Noack BR. 2016 *Machine learning control: taming nonlinear dynamics and turbulence*. Berlin, Germany: Springer.
12. Majda AJ, Lee Y. 2014 Conceptual dynamical models for turbulence. *Proc. Natl Acad. Sci. USA* **111**, 6548–6553. (doi:10.1073/pnas.1404914111)
13. Schaeffer H. 2017 Learning partial differential equations via data discovery and sparse optimization. *Proc. R. Soc. A* **473**, 20160446. (doi:10.1098/rspa.2016.0446)
14. Farazmand M, Sapsis TP. 2016 Dynamical indicators for the prediction of bursting phenomena in high-dimensional systems. *Phys. Rev. E* **94**, 032212. (doi:10.1103/PhysRevE.94.032212)
15. Bongard J, Lipson H. 2007 Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **104**, 9943–9948. (doi:10.1073/pnas.0609476104)
16. Einicke GA, White LB. 1999 Robust extended Kalman filtering. *IEEE Trans. Signal Process.* **47**, 2596–2599. (doi:10.1109/78.782219)
17. Julier SJ, Uhlmann JK. 1997 A new extension of the Kalman filter to nonlinear systems. *Proc. SPIE* **3068**, 182–193. (doi:10.1117/12.280797)
18. Lee Y, Majda AJ. 2016 State estimation and prediction using clustered particle filters. *Proc. Natl Acad. Sci. USA* **113**, 14 609–14 614. (doi:10.1073/pnas.1617398113)
19. Comeau D, Zhao Z, Giannakis D, Majda AJ. 2017 Data-driven prediction strategies for low-frequency patterns of North Pacific climate variability. *Clim. Dyn.* **48**, 1855–1872. (doi:10.1007/s00382-016-3177-5)
20. Tatsis K, Dertimanis V, Abdallah I, Chatzi E. 2017 A substructure approach for fatigue assessment on wind turbine support structures using output-only measurements. *Procedia Engineering* **199**, 1044–1049. (doi:10.1016/j.proeng.2017.09.285)
21. Quade M, Abel M, Shafi K, Niven RK, Noack BR. 2016 Prediction of dynamical systems by symbolic regression. *Phys. Rev. E.* **94**, 012214. (doi:10.1103/PhysRevE.94.012214)
22. Mirmomeni M, Lucas C, Moshiri B, Araabi NB. 2010 Introducing adaptive neurofuzzy modeling with online learning method for prediction of time-varying solar and geomagnetic activity indices. *Expert Syst. Appl.* **37**, 8267–8277. (doi:10.1016/j.eswa.2010.05.059)
23. Gholipour A, Lucas C, Araabi NB, Mirmomeni M, Shafiee M. 2007 Extracting the main patterns of natural time series for long-term neurofuzzy prediction. *Neural Comput. Appl.* **16**, 383–393. (doi:10.1007/s00521-006-0062-x)
24. Mirmomeni M, Lucas C, Araabi NB, Moshiri B, Bidar MR. 2011 Recursive spectral analysis of natural time series based on eigenvector matrix perturbation for online applications. *IET Signal Process.* **5**, 512–526. (doi:10.1049/iet-spr.2009.0278)
25. Mirmomeni M, Lucas C, Araabi NB, Moshiri B, Bidar MR. 2011 Online multi-step ahead prediction of time-varying solar and geomagnetic activity indices via adaptive neurofuzzy modeling and recursive spectral analysis. *Sol. Phys.* **272**, 189–213. (doi:10.1007/s11207-011-9810-x)
26. Marques CAF, Ferreira JA, Rocha A, Castanheira JM, Melo-Goncalves P, Vaz N, Dias JM. 2006 Singular spectrum analysis and forecasting of hydrological time series. *Phys. Chem. Earth, Parts A/B/C* **31**, 1172–1179. (doi:10.1016/j.pce.2006.02.061)
27. Abdollahzade M, Miranian A, Hassani H, Iranmanesh H. 2015 A new hybrid enhanced local linear neuro-fuzzy model based on the optimized singular spectrum analysis and its

application for nonlinear and chaotic time series forecasting. *Inf. Sci.* (*Ny*) **295**, 107–125. (doi:10.1016/j.ins.2014.09.002)
28. Ye L, Liu P. 2011 Combined model based on EMD-SVM for short-term wind power prediction. *Commun. Nonlinear Science Numer. Simul.* **31**, 102–108.
29. Cousins W, Sapsis TP. 2014 Quantification and prediction of extreme events in a one-dimensional nonlinear dispersive wave model. *Phys. D* **280–281**, 48–58. (doi:10.1016/j.physd.2014.04.012)
30. Cousins W, Sapsis TP. 2016 Reduced order precursors of rare events in unidirectional nonlinear water waves. *J. Fluid Mech.* **790**, 368–388. (doi:10.1017/jfm.2016.13. 368)
31. Lorenz EN. 1969 Atmospheric predictability as revealed by naturally occurring analogues. *J. Atmos. Sci.* **26**, 636–646. (doi:10.1175/1520-0469(1969)26<636:APARBN>2.0.CO;2)
32. Xavier PK, Goswami BN. 2007 An analog method for real-time forecasting of summer monsoon subseasonal variability. *Mon. Weather Rev.* **135**, 4149–4160. (doi:10.1175/2007MWR1854.1)
33. Zhao Z, Giannakis D. 2016 Analog forecasting with dynamics-adapted kernels. *Nonlinearity* **29**, 2888–2939. (doi:10.1088/0951-7715/29/9/2888)
34. Chiavazzo E, Gear CW, Dsilva CJ, Rabin N, Kevrekidis IG. 2014 Reduced models in chemical kinetics via nonlinear data-mining. *Processes* **2**, 112–140. (doi:10.3390/pr2010112)
35. Wan ZY, Sapsis TP. 2017 Reduced-space Gaussian Process Regression for data-driven probabilistic forecast of chaotic dynamical systems. *Phys. D: Nonlinear Phenom.* **345**, 45–55. (doi:10.1016/j.physd.2016.12.005)
36. Rasmussen CE, Williams CKI. 2006 *Gaussian processes for machine learning*. Cambridge, MA: The MIT Press.
37. Hochreiter S, Schmidhuber J. 1997 Long short-term memory. *Neural. Comput.* **9**, 1735–1780. (doi:10.1162/neco.1997.9.8.1735)
38. Gers FA, Eck D, Schmidhuber J. 2001 Applying LSTM to time series predictable through time-window approaches. In *Proc. Int. Conf. on Artificial Neural Networks, ICANN'01 Vienna, 21–25 August* (eds G Dorffner, H Bischof, K Hornik), pp. 669–676. Lecture Notes in Computer Science, vol. 2130. New York, NY: Springer.
39. Rico-Martínez R, Krischer K, Kevrekidis IG, Kube MC, Hudson JL. 1992 Discrete- versus continuous-time nonlinear signal processing of Cu electrodissolution data. *Chem. Eng. Commun.* **118**, 25–48. (doi:10.1080/00986449208936084)
40. Jaeger M, Haas H. 2004 Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78–80. (doi:10.1126/science.1091277)
41. Chatzis SP, Demiris Y. 2011 Echo state Gaussian process. *IEEE Trans. Neural Netw.* **22**, 1435–1445. (doi:10.1109/TNN.2011.2162109)
42. Broomhead DS, Lowe D. 1988 Multivariable functional interpolation and adaptive networks. *Complex Syst.* **2**, 321–355. (doi:10.4236/jbise.2013.65A003)
43. Kim KB, Park JB, Choi YH, Chen G. 2000 Control of chaotic dynamical systems using radial basis function network approximators. *Inf. Sci.* **130**, 165–183. (doi:10.1016/S0020-0255(00)00074-8)
44. Pathak J, Hunt BR, Girvan M, Lu Z, Ott E. 2018 Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. *Phys. Rev. Lett.* **120**, 024102. (doi:10.1103/PhysRevLett.120.024102)
45. Pathak J, Lu Z, Hunt BR, Girvan M, Ott E. 2017 Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data. *Chaos.* **27**, 121102. (doi:10.1063/1.5010300)
46. Takens F. 1981 Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick, UK, 1980*, pp. 366–381. Lecture Notes in Mathematics, vol. 898. Berlin, Germany: Springer.
47. Hochreiter J. 1991 Untersuchungen zu dynamischen neuronalen Netzen. Master thesis, Institut fur Informatik, Technische Universitat, Munchen.
48. Bengio Y, Simard P, Frasconi P. 1994 Long short-term memory. *IEEE Trans. Neural Netw.* **5**, 157–166. (doi:10.1109/72.279181)
49. Wierstra D, Schmidhuber J, Gomez FJ. 2005 Evolino: hybrid neuroevolution/optimal linear search for sequence learning. *Proc. 19th IJCAI*, 853–858.
50. Graves A, Mohamed AR, Hinton G. 2013 Speech recognition with deep recurrent neural networks. In *Proc. Acoustic, Speech and Signal Processing, ICASSP, Vancouver, Canada, 26–31 May*, pp. 6645–6649. Piscataway, NJ: IEEE.

51. Graves A, Fernández S, Liwicki M, Bunke H, Schmidhuber J. 2007 Unconstrained online handwriting recognition with recurrent neural networks. In *Proc. 20th NIPS, Vancouver, Canada, 3–6 December*, pp. 577–584. Curran Associates.
52. Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q. 2016 Google's neural machine translation system: bridging the gap between human and machine translation. (http://arxiv.org/abs/1609.08144)
53. Kingma DP, Ba J. 2017 Adam: a method for stochastic optimization. (http://arxiv.org/abs/1412.6980)
54. Glorot X, Bengio Y. 2010 Understanding the difficulty of training deep feedforward neural networks. *Proc. 13th AISTATS* **114**, 249–256.
55. Majda A, Harlim J. 2012 *Filtering complex turbulent systems*. Cambridge, UK: Cambridge University Press.
56. Majda A, Abramov RV, Grote MJ. 2005 *Information theory and stochastics for multiscale nonlinear systems*. Centre de Recherches Mathematiques Monograph Series, vol. 25. Providence, RI: American Mathematical Society.
57. Allgaier NA, Harris KD, Danforth CM. 2012 Empirical correction of a toy climate model. *Phys. Rev. E* **85**, 026201. (doi:10.1103/PhysRevE.85.026201)
58. Basnarkov L, Kocarev L. 2012 Forecast improvement in Lorenz96 system. *Nonlinear Process. Geophys.* **19**, 569–575. (doi:10.5194/npg-19-569-2012)
59. Lorenz NE. 1996 Predictability: a problem partly solved. In *Proc. Seminar held at ECMWF on Predictability. Reading, UK, 4–8 September*, pp. 1–18.
60. Crommelin DT, Majda AJ. 2004 Strategies for model reduction: comparing different optimal bases. *J. Atmos. Sci.* **61**, 2206–2217. (doi:10.1175/1520-0469(2004)061<2206:SFMRCD>2.0.CO;2)
61. Kuramoto Y, Tsuzuki T. 1976 Persistent propagation of concentration waves in dissipative media far from thermal equilibrium. *Prog. Theor. Phys.* **55**, 356–369. (doi:10.1143/PTP.55.356)
62. Kuramoto Y. 1978 Diffusion-induced chaos in reaction systems. *Prog. Theor. Phys. Suppl.* **64**, 346–367. (doi:10.1143/PTPS.64.346)
63. Sivashinsky G. 1977 Nonlinear analysis of hydrodynamic instability in laminar flames–I. Derivation of basic equations. *Acta. Astronaut.* **4**, 1177–1206. (doi:10.1016/0094-5765(77)90096-0)
64. Sivashinsky G, Michelson DM. 1980 On irregular wavy flow of a liquid film down a vertical plane. *Prog. Theor. Phys.* **63**, 2112–2114. (doi:10.1143/PTP.63.2112)
65. Blonigan PJ, Wang Q. 2014 Least squares shadowing sensitivity analysis of a modified Kuramoto-Sivashinsky equation. *Chaos Solitons Fractals* **64**, 16–25. (doi:10.1016/j.chaos.2014.03.005)
66. Kevrekidis IG, Nicolaenko B, Scovel JC. 1990 Back in the saddle again: a computer assisted study of the kuramoto-sivashinsky equation. *SIAM J. Appl. Math.* **50**, 760–790. (doi:10.1137/0150045)
67. Selten FM. 1995 An efficient description of the dynamics of barotropic flow. *J. Atmos. Sci.* **52**, 915–936. (doi:10.1175/1520-0469(1995)052<0915:AEDOTD>2.0.CO;2)
68. Thompson DWJ, Wallace JM. 2000 Annular modes in the extratropical circulation: Part I: Month-to-month variability. *J. Clim.* **13**, 1000–1016. (doi:10.1175/1520-0442(2000)013<1000:AMITEC>2.0.CO;2)
69. Mo KC, Livezey RE. 1986 Tropical-extratropical geopotential height teleconnections during the northern hemisphere winter. *Mon. Weather Rev.* **114**, 2488–2512. (doi:10.1175/1520-0493(1986)114%3C2488:TEGHTD%3E2.0.CO;2)

Appendix: Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks

Pantelis R. Vlachas*, Wonmin Byeon*, Zhong Y. Wan†, Themistoklis P. Sapsis†, Petros Koumoutsakos*

May 10, 2018

A Long short-term memory (LSTM)

A.1 Training and inference

In this section, the LSTM training procedure is explained in detail. We assume that time series data stemming from a dynamical system is available in the form $D = \{z_{t:N}, \dot{z}_{t:N}\}$, where $z_t \in \mathbb{R}^{d_i}$ is the state at time step t and $\dot{z}_t$ is the derivative. The available time series data are divided into two separate sets, the training dataset and the validation dataset, i.e. z_t^{train}, $\dot{z}_t^{train}$, $t \in \{1, \cdots, N_{train}\}$, and z_t^{val}, $\dot{z}_t^{val}$, $t \in \{1, \cdots, N_{val}\}$. N_{train} and N_{val} are the number of training and validation samples respectively. We set the ratio to $N_{train}/N = 0.8$. This data is stacked as

$$\underbrace{\mathbf{i}_t^{train} = \begin{pmatrix} z_{t+d-1}^{train} \\ z_{t+d-2}^{train} \\ \vdots \\ z_t^{train} \end{pmatrix}}_{\text{Input stack}}, \quad \underbrace{\mathbf{o}_t^{train} = \dot{z}_{t+d-1}^{train}}_{\text{Output stack}}, \tag{1}$$

for $t \in \{1, 2, \ldots, N_{train} - d + 1\}$, in order to form the training (and validation) input and output of the LSTM. These training samples are used to optimize the parameters of the LSTM (weights and biases) in order to learn the mapping $\mathbf{i}_t \rightarrow \mathbf{o}_t$. The loss function of each sample is

$$\mathcal{L}_{sample}(\mathbf{i}_t^{train}, \mathbf{o}_t^{train}, w) = ||\mathcal{F}^w(\underbrace{z_{t:t-d+1}^{train}}_{\mathbf{i}_t^{train}}) - \mathbf{o}_t^{train}||^2, \tag{2}$$

while the total Loss is defined as

$$\mathcal{L}(D, w) = \frac{1}{S} \sum_{b=1}^{S} \mathcal{L}(\mathbf{i}_b^{train}, \mathbf{o}_b^{train}, w), \tag{3}$$

*Chair of Computational Science, ETH Zurich, Clausiusstrasse 33, Zurich, CH-8092, Switzerland

†Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, United States

where $S = N_{train} - d + 1$ is the total number of samples. These samples can be further stacked together as batches of size B, with the loss of the batch defined as the mean loss of the samples belonging to the batch. Using only one sample for the loss gradient estimation may lead to noisy gradient estimates and slow convergence. Mini-batch optimization tackles this issue.

At the beginning of the training the weights are randomly initialized to w^0 using Xavier initialization. We also tried other initialization methods like drawing initial weights from random normal distributions, or initializing them to constant values, but they often led to saturation of the activation functions, especially for architectures with higher back-propagation horizon d. The training proceeds by optimizing the network weights iteratively for each batch. In order to perform this optimization step, a gradient descent optimizer can be used

$$w^{i+1} = w^i - \eta \nabla_w \mathcal{L}(\mathbf{i}_t^{train}, \mathbf{o}_t^{train}, w^i), \tag{4}$$

where η is the step-size parameter, w^i are the weights before optimizing the batch i and w^{i+1} are the updated weights. Plain gradient descent optimization suffers from slow convergence in practice and convergence to local sub-optimal solutions. This approach is especially not well-suited for high dimensional problems in deep learning where the number of parameters (weights) to be optimized lie in a high-dimensional manifold with many local optima. Sparse gradients stemming from the mini-batch-optimization lead also to slow convergence as previously computed gradients are ignored. Recent advances in stochastic optimization led to the invention of adaptive schemes that efficiently cope with the aforementioned problems.

In our work, we used the Adam stochastic optimization method. Adam exploits previously computed gradients using moments. The weights are initialized to w^0 and the moment vectors to m_1^0 and m_2^0. At each step the updates in the Adam optimizer are

$$\begin{aligned}
g &= \nabla_w \mathcal{L}(\mathbf{i}_t^{train}, \mathbf{o}_t^{train}, w^i) \\
m_1^{i+1} &= \beta_1 m_1^i + (1 - \beta_1)\, g \\
m_2^{i+1} &= \beta_2 m_2^i + (1 - \beta_2)\, g^2 \\
\hat{m}_1 &= m_1^{i+1}/(1 - \beta_1^i) \\
\hat{m}_2 &= m_2^{i+1}/(1 - \beta_2^i) \\
w_{i+1} &= w_i - \eta\, \hat{m}_1/(\sqrt{\hat{m}_2} + \epsilon),
\end{aligned} \tag{5}$$

where $\beta_1, \beta_2, \epsilon$, and η are hyper-parameters, g^2 is the point-wise square of the gradient and β_1^i is the parameter β_1 in the ith power, where i is the iteration number. After updating the weights using the Adam procedure (5) for every batch, a *training epoch* is completed. Many such epochs are performed until the total training loss reaches a plateau. After each epoch the loss is evaluated also in the validation data set, in order to avoid overfitting. The validation loss is used as a proxy of the generalization error. The training is stopped when the validation error is not decreasing for 30 consecutive epochs or the maximum of 1000 epochs is reached. In our work we used $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$. We found that our results were robust towards the selection of these hyper-parameters. To speed up convergence speed, a higher initial learning rate $\eta = 0.001$ was used and the models were then refined with $\eta = 0.0001$.

A.2 Weighting the loss function

In the training procedure described above the loss function for each sample is given by

$$\mathcal{L}_{sample}(\mathbf{i}_t, \mathbf{o}_t, w) = ||\mathcal{F}^w(\underbrace{z_{t:t-d+1}}_{\mathbf{i}_t}) - \mathbf{o}_t||^2. \tag{6}$$

However, in the applications considered in this paper the neural network output $\mathcal{F}^w$ is a multidimensional vector and represents a prediction of the derivative of the reduced order state of a dynamical system. In a dynamical system, specific reduced order states are more important than others as they may explain a bigger portion of the total energy. This importance can be introduced in the loss function by assigning different weights in different outputs of the neural network. The loss of each sample takes then the following form

$$\mathcal{L}_{sample}(\mathbf{i}_t^j, \mathbf{o}_t^j, w) = \frac{1}{d_o}\sum_{j=1}^{d_o} w_j \Big(\mathcal{F}^w(\underbrace{z_{t:t-d+1}^j}_{\mathbf{i}_t^j}) - \mathbf{o}_t^j\Big)^2, \tag{7}$$

where d_o is the output dimension and weights w_j are selected according to the significance of each output component, e.g. energy of each component in the physical system.

A.3 LSTM architecture

An RNN unfolded d temporal time steps in the past is illustrated in Figure 1. The following discussion on *Stateless* and *Stateful* RNNs generalizes to LSTMs, with the only difference that the hidden state consists of $\mathbf{h}_t, C_t$ instead of solely $\mathbf{h}_t$ and the functions coupling the hidden states with the input as well as the output with the hidden states are more complicated.

In *Stateless* RNNs the hidden states at the truncation layer d, $\mathbf{h}_{t-d}$ are initialized always with 0. As a consequence, $\mathbf{o} = \mathcal{F}^w(\mathbf{i}_{t:t-d+1})$ and only the short-term history is used to perform a prediction. The only difference when using LSTM cells is that the function $\mathcal{F}^w$ has a more complex structure and additionally $\mathbf{h}_{t-d}, C_{t-d} = 0$.

In contrast, in *Stateful* RNNs the states $\mathbf{h}_{t-d} \neq 0$. In this case, these states can be initialized by "teacher forcing" the RNN using data from a longer history in the past. For example, assuming $\mathbf{i}_{t-d:t-2d+1}$ is known, we can set $\mathbf{h}_{t-2d} = 0$, and compute $\mathbf{h}_{t-d}$ using the given history $\mathbf{i}_{t-d:t-2d+1}$ ignoring the outputs. This value can then be used to predict $\mathbf{o}_t = \mathcal{F}^w(\mathbf{i}_{t:t-d+1}, \mathbf{h}_{t-d})$ as in Figure 1. This approach has two disadvantages.

- In order to be able to forecast starting from various initial conditions, even with "teacher forcing" some initialization of the hidden states is imperative. This initialization introduces additional error, which is not the case for the *Stateless* RNN.

- In the *Stateful* RNN a longer history needs to be known in order to initialize the hidden states with "teacher forcing". Even though more data needs to be available, we did not observe any prediction accuracy improvement in the cases considered. This follows from the chaotic nature

of the systems, as information longer than some time-steps in the past are irrelevant for the prediction.

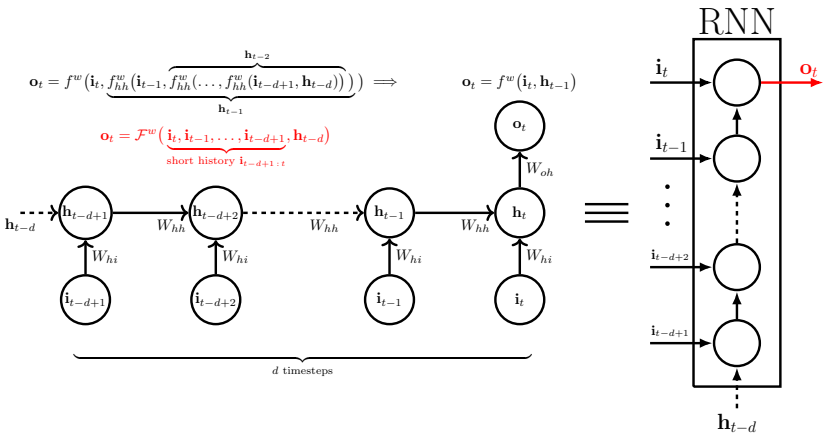

Figure 1: An RNN unfolded d timesteps. In mathematical terms, unfolding is equivalent with iteratively applying f^w_{hh} to $\mathbf{h}_{t-d}$ and finally feeding the result to the output function f^w. The output of the RNN is thus a function of the d previous inputs $\mathbf{i}_{t:t-d+1}$ and the initialization of the hidden states $\mathbf{h}_{t-d}$. This function is denoted with $\mathcal{F}^w$. For the RNN the hidden state mapping has the simple form $f^w_{hh}(\mathbf{i}_t, \mathbf{h}_{t-1}) = \sigma_h(W_{hi}\mathbf{i}_t + W_{hh}\mathbf{h}_{t-1})$, while the output mapping is $f^w(\mathbf{i}_t, \mathbf{h}_{t-1}) = \sigma_o(W_{oh}\mathbf{h}_t) = \sigma_o\big(W_{oh} f^w_{hh}(\mathbf{i}_t, \mathbf{h}_{t-1})\big)$. The same argumentation holds for LSTM, though the form of f^w_{hh}, f^w and $\mathcal{F}^w$ are more complicated.

B Lorenz 96

The most energetic Fourier modes in the Lorenz 96 system for different forcing regimes $F \in \{4, 6, 8, 16\}$ are given in Table 1. These modes are used in order to construct the reduced order phase space.

Forcing	Wavenumbers k	Forcing	Wavenumbers k
$F = 4$	7,10,14,9,17,16	$F = 8$	8,9,7,10,11,6
$F = 6$	8,7,9,10,11,6	$F = 16$	8,9,10,7,11,6

Table 1: Most energetic Fourier modes used in the reduced order phase space

C Kuramoto-Sivashinsky equation

C.1 Dimensionality reduction

The temporal average of the state of the Kuramoto-Sivashinsky equation and the cumulative energy are plotted in Figure 2. As ν declines, chaoticity in the

system rises and higher oscillations of the mean towards the Dirichlet boundary conditions are observed in Figure 2, while the number of modes needed to capture most of the energy is higher.

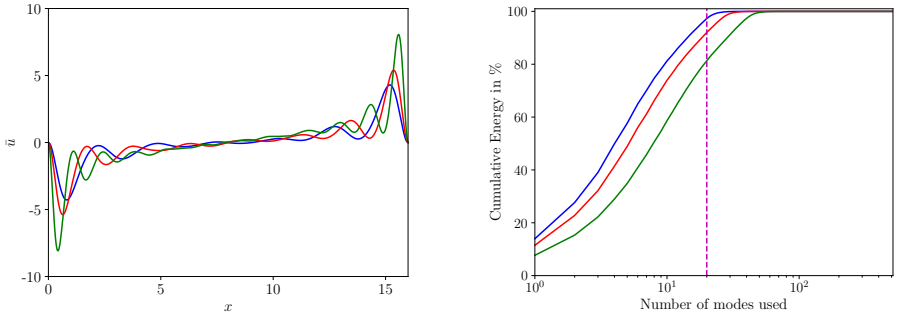

Figure 2: Temporal average $\overline{\mathbf{u}}$ and cumulative mode (PCA) energy for different values of ν in the Kuramoto-Sivashinsky system.
$1/\nu = 10$ ——; $1/\nu = 16$ ——; $1/\nu = 36$ ——; 20 modes - - -

C.2 Results

For the hybrid LSTM-MSM, the ratio of the ensemble members that are modeled with LSTM is plotted with respect to time in Figure 3a. The quotient drops slower for $1/\nu = 10$ in the long run as the intrinsic dimensionality of the attractor is smaller and trajectories diverge slower. However, in the beginning the LSTM ratio is higher for $1/\nu = 16$ as the MSM drives initial conditions close to the boundary faster towards the attractor due to the higher damping coefficients compared to the case $1/\nu = 10$. This explains the initial knick in the graph for $1/\nu = 16$. The slow damping coefficients for $1/\nu = 10$ do not allow the MSM to drive the trajectories back to the attractor in a faster pace than the diffusion caused by the LSTM forecasting. Compared with GPR plotted in Figure 3b, the ratio drops slower.

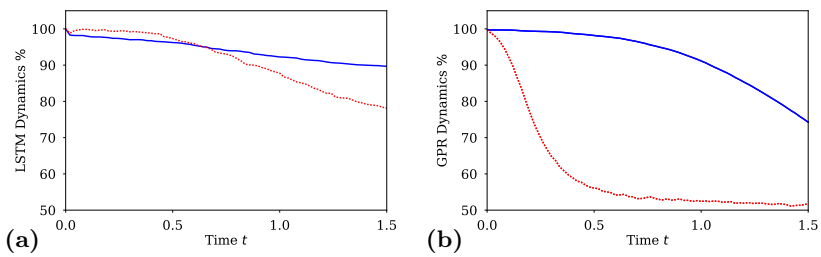

Figure 3: **(a)** Ratio of LSTM-MSM ensemble members modeled by the LSTM dynamics for the Kuramoto-Sivashinsky ($T = 1.5$). **(b)** The same for GPR in the hybrid GPR-MSM. (Mean over 1000 initial conditions)
$1/\nu = 10$ ——; $1/\nu = 16$ ·····

D Barotropic model

In this section we describe the method used to reduce the dimensionality of the Barotropic climate model. First, the original problem dimension of 231 is reduced using a generalized version of the classical multidimensional scaling method. Then, we construct Empirical Orthogonal Functions (EOFs) that form an orthogonal basis of the reduced order space and project the dynamics to them.

The classical multidimensional scaling procedure tries to identify an embedding with a lower dimensionality such that the pairwise inner products of the dataset are preserved. Assuming that the dataset consists of points ζ_i, $i \in \{1, \dots, N\}$, whose reduced order representation is denoted with $\mathbf{y}_i$, the procedure is equivalent with the solution of the following optimization problem

$$\underset{\mathbf{y}_1,\dots,\mathbf{y}_N}{\text{minimize}} \sum_{i<j} \left(\langle \zeta_i, \zeta_j \rangle_\zeta - \langle \mathbf{y}_i, \mathbf{y}_j \rangle_\mathbf{y} \right)^2, \tag{8}$$

where $\langle \cdot, \cdot \rangle_\zeta$, and $\langle \cdot, \cdot \rangle_\mathbf{y}$ denote some well defined inner product of the original space ζ and the embedding space $\mathbf{y}$ respectively. Problem (8) minimizes the total squared error between pairwise products. In case both products are the scalar products, the solution of (8) is equivalent with PCA. Assuming only $\langle \cdot, \cdot \rangle_\mathbf{y}$ is the scalar product, problem (8) also accepts an analytic solution. Let $W_{ij} = \langle \zeta_i, \zeta_j \rangle_\zeta$ be the coefficients of the Gram matrix, $|k_1| \geq |k_2| \geq \cdots \geq |k_N|$ its eigenvalues sorted in descending absolute value and $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_N$ the respective eigenvectors. The optimal d-dimensional embedding for a point ζ_n is given by

$$\mathbf{y}_n = \begin{pmatrix} k_1^{1/2} \mathbf{u}_1^n \\ k_2^{1/2} \mathbf{u}_2^n \\ \vdots \\ k_d^{1/2} \mathbf{u}_d^n \end{pmatrix}, \tag{9}$$

where $\mathbf{u}_m^n$ denotes the nth component of the mth eigenvector. The optimality of (9) can be proven by the Eckart-Young-Mirsky theorem, as problem (8) is equivalent with finding the best d rank approximation in the Frobenius norm. In our problem, the standard kinetic energy product is used to preserve the nonlinear symmetries of the system dynamics:

$$\langle \zeta_i, \zeta_j \rangle_\zeta = \int_\mathcal{S} \nabla \psi_i \cdot \nabla \psi_j \, d\mathcal{S} = -\int_\mathcal{S} \zeta_i \psi_j \, d\mathcal{S} = -\int_\mathcal{S} \zeta_j \psi_i \, d\mathcal{S}, \tag{10}$$

where the last identities are derived using partial integration and the fact that $\zeta = \Delta \mathbf{y}$.

Solution (9) is only optimal w.r.t. the N training data points used to construct the Gram matrix. In order to calculate the embedding for a new point, it is convenient to compute the EOFs which form an orthogonal basis of the reduced order space $\mathbf{y}$. The EOFs are given by

$$\phi_m = \sum_{n=1}^{N} k_m^{-1/2} \mathbf{u}_m^n \zeta_n, \tag{11}$$

where m runs from 1 to d. The EOFs are sorted in descending order according to their energy level. The first four EOFs are plotted in Figure 4. EOF analysis

has been used to identify individual realistic climatic modes such as the Arctic Oscillation (AO) known as `teleconnections`. The first EOF is characterized by a center of action over the Arctic that is surrounded by a zonal symmetric structure in mid-latitudes. This pattern resembles the Arctic Oscillation/Northern Hemisphere Annular Mode (AO/NAM) and explains approximately 13.5% of the total energy. The second, third and fourth EOFs are quantitatively very similar to the East Atlantic/West Russia, the Pacific/North America (PNA) and the Tropical/Northern Hemisphere (TNH) patterns end account for 11.4%, 10.4% and 7.1% of the total energy respectively. Since these EOFs feature realistic climate teleconnections, performing accurate predictions of them is of high practical importance.

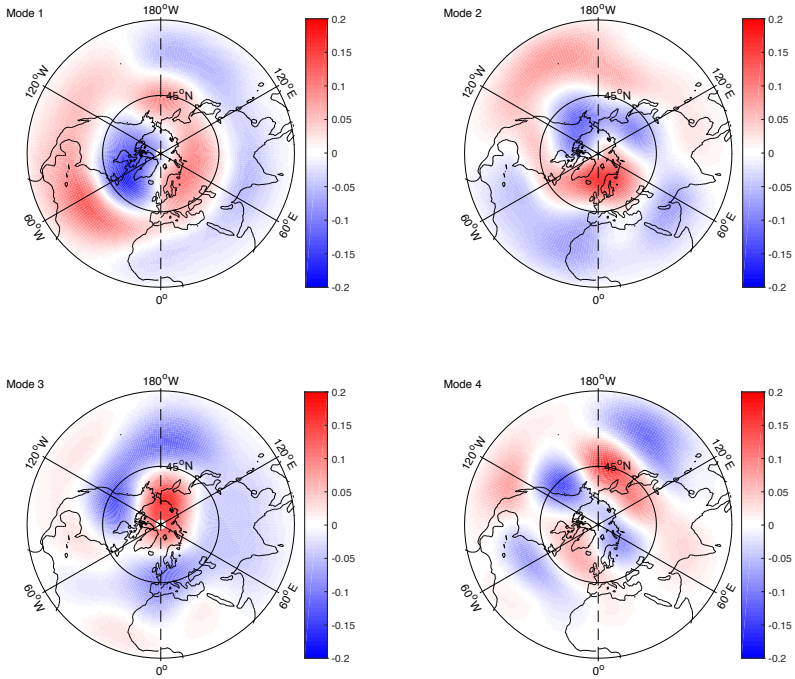

Figure 4: The four most energetic empirical orthogonal functions of the barotropic model

As a consequence of the orthogonality of the EOFs w.r.t. the kinetic energy product, the reduced representation $\mathbf{y}^*$ of a new state ζ^* can be recovered from

$$\mathbf{y}^* = \begin{pmatrix} \langle \zeta^*, \phi_1 \rangle_\zeta \\ \langle \zeta^*, \phi_2 \rangle_\zeta \\ \vdots \\ \langle \zeta^*, \phi_d \rangle_\zeta \end{pmatrix}. \tag{12}$$

Note that only the d coefficients corresponding to the most energetic EOFs form the reduced order state $\mathbf{y}^*$. In essence, the EOFs act as an orthogonal basis of the reduced order space and the state obtained from classical multidimensional scaling ζ^* is projected to this basis.

E Sensitivity to noise in the data

In this section, we evaluate the robustness of the proposed approach to noise. For this purpose, the training data are perturbed with different noise levels. We add Gaussian noise sampled from $N(0, \sigma_{noise})$ where the noise standard deviation is proportional to the attractor standard deviation $\sigma_{attractor}$ of each system, i.e. $\sigma_{noise} = k\,\sigma_{attractor}$. We note that $\sigma_{attractor}$ is computed from the training data, as the standard deviation of the samples of the reduced order state of the system. Different noise levels k are considered.

E.1 Lorenz 96

In the following, we analyze the influence of noise in the training data for the Lorenz 96 system. In parallel with the main body of the paper, we plot the RMSE error evolution of the most energetic mode (first row of Figure 5) for short term till $T = 0.1$, the same for time $T = 2$ (second row of Figure 5) and the ACC (third row of Figure 5). The columns of Figure 5 correspond to different chaotic regimes of the Lorenz 96 system. For the forcing $F = 4$ and noise levels $k \in \{0.01, 0.2\}$, noise does not affect the prediction performance of the LSTM. This can be attributed to the fact that the attractor dimensionality is really low in this case and the amount of data is enough to capture the dynamics despite the noisy training data. However, for $F = 8$ and $F = 16$ adding noise leads to slight deterioration of the short term prediction accuracy for the noise level $k = 0.01$, as illustrated by the last two Figures in the first row of Figure 5. As a consequence, the method can be considered robust against noise. Increasing the noise level to $k = 0.2$ corresponding to a noise standard deviation equal to 20% of the attractor standard deviation leads to an important deterioration in short term prediction performance. The deterioration in the short term prediction performance can be seen in the short term RMSE error evolution of the fourth most energetic modes for the forcing regime $F = 8$ plotted in Figure 6.

E.2 Kuramoto-Sivashinsky equation

In Figure 7 we plot the RMSE error evolution for the most energetic mode and the ACC of the Kuramoto-Sivashinsky equation for two different chaotic regimes $1/\nu \in \{10, 16\}$. Three different noise levels $k \in \{0.001, 0.01, 0.2\}$ are considered. For the low chaotic regime $1/\nu = 10$, predictability performance is robust against noise, as the error evolution changes slightly with $k \in \{0.001, 0.01\}$. Only when the training data are polluted with noise with a standard deviation bigger than 20% of the attractor standard deviation is the predictability performance greatly deteriorated. On the contrary, adding noise to the training data in the input improves the predictability performance of LSTM for the chaotic regime $\nu = 1/16$. This can be attributed to the fact that in this chaotic regime correlation patterns are much less prominent, and the LSTM is more prone to overfit. As a consequence, adding noise to the input forces the neural network to learn only robust patterns in the data that can be generalized. Short term prediction performance is deteriorated slightly, but in the long term, the LSTM is more robust against accumulation of errors. This behavior has to be further investigated in future work.

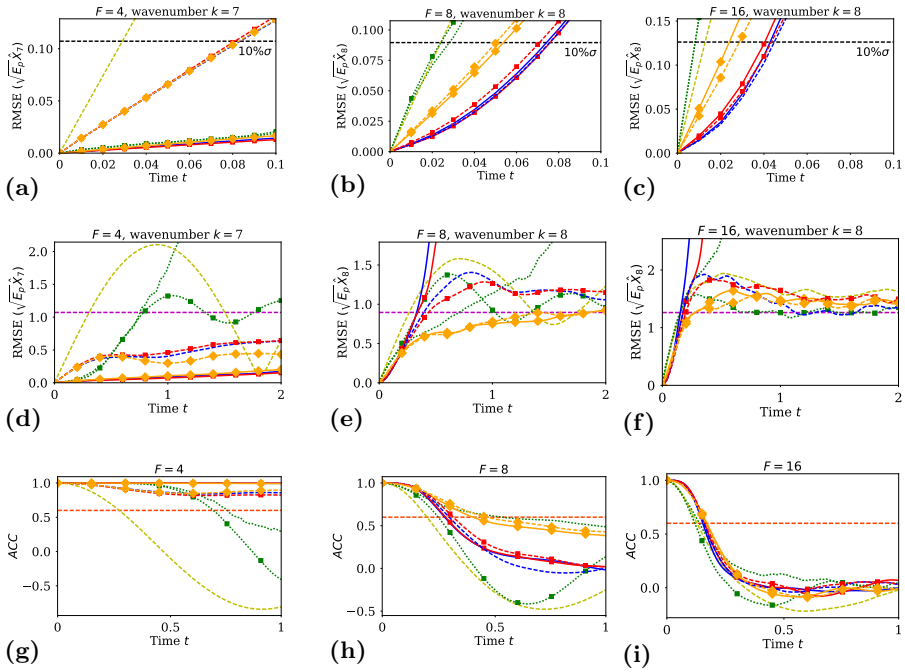

Figure 5: **(a), (b), (c)** Short term RMSE evolution of the most energetic mode for forcing regimes $F = 4, 8, 16$ respectively of the Lorenz 96 system. **(d), (e), (f)** Long term RMSE evolution. **(g), (h), (i)** Evolution of the ACC coefficient. (In all plots average over 1000 initial conditions is reported).
10% $\sigma_{attractor}$ - - -; $\sigma_{attractor}$ - - -; $ACC = 0.6$ threshold - - -; MSM- - -; GPR·····; GPR-MSM·■·; LSTM $k = 0‰$——; LSTM-MSM $k = 0‰$- - -;
LSTM $k = 10‰$-■-; LSTM-MSM $k = 10‰$-■-; LSTM $k = 200‰$-◆-;
LSTM-MSM $k = 200‰$-◆-

E.3 Barotropic model

In Figure 8 we plot the RMSE error evolution for the four most energetic EOFs of the Barotropic model. Three different noise levels $k \in \{0.001, 0.01, 0.2\}$ are considered. Only for the highest noise level is the prediction performance deteriorated. For low noise levels, the prediction performance can be increased ($k = 0.001$), as the noise may regularize the Back-propagation procedure during training with stochastic methods. Adding noise to the input of neural networks can be used as a practical heuristic to increase their accuracy and can also be seen as a form of dropout in the input layer of the LSTM. The results indicate that the prediction performance of the LSTM is robust for the noise levels $k \in \{0.001, 0.01\}$.

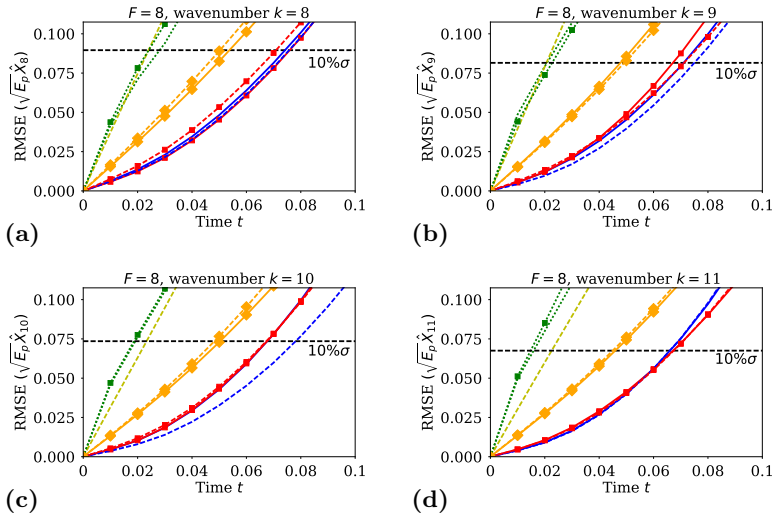

Figure 6: RMSE prediction error evolution of four energetic modes for the Lorenz 96 system with forcing $F = 8$. **(a)** Most energetic mode $k = 8$. **(b)** Low energy mode $k = 9$. **(c)** Low energy mode $k = 10$. **(d)** Low energy mode $k = 11$. (In all plots average over 1000 initial conditions reported)
10% $\sigma_{attractor}$ - - -; MSM- - -; GPR·····; GPR-MSM··■·; LSTM $k = 0$‰——; LSTM-MSM $k = 0$‰- - -; LSTM $k = 10$‰-■—; LSTM-MSM $k = 10$‰-■-; LSTM $k = 200$‰-◆—; LSTM-MSM $k = 200$‰-◆-

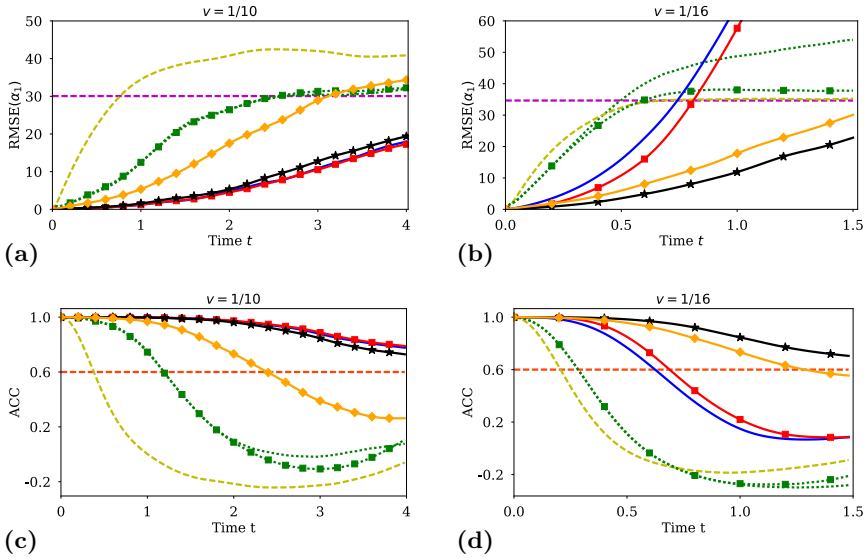

Figure 7: Training data of LSTM are perturbed with standard deviation $\sigma_{noise} = k\,\sigma_{attractor}$. Three different noise levels $k \in \{0.001, 0.01, 0.2\}$ are considered. **(a)**, **(b)** RMSE evolution of the most energetic mode of the K-S equation with $1/\nu = 10$ and $1/\nu = 16$. **(c)**, **(d)** ACC evolution of the most energetic mode of the K-S equation with $1/\nu = 10$ and $1/\nu = 16$. (In all plots, average value over 1000 initial conditions is reported)
$\sigma_{attractor}$; $ACC = 0.6$ threshold; MSM; GPR; GPR-MSM; LSTM $k = 0‰$; LSTM $k = 1‰$; LSTM $k = 10‰$; LSTM $k = 200‰$

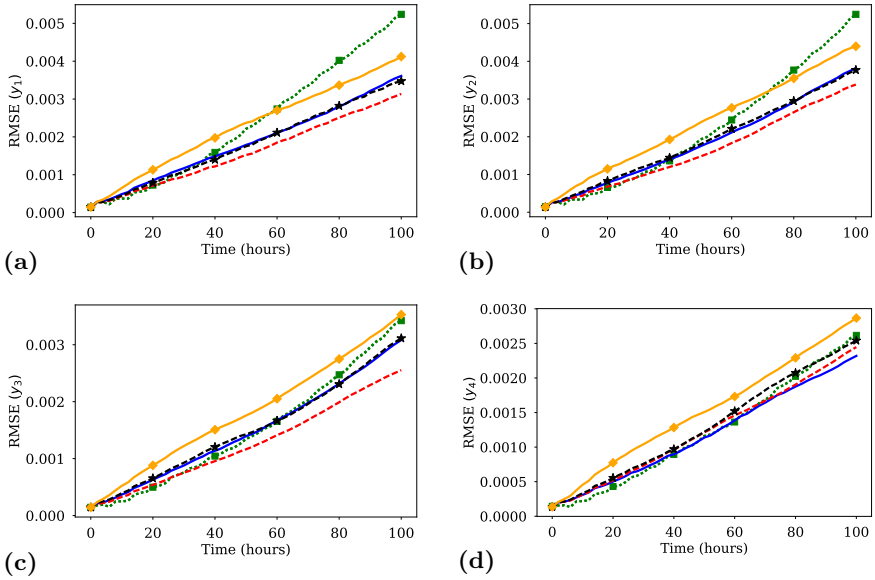

Figure 8: RMSE evolution of the four most energetic EOFs for the Barotropic climate model, average over 500 initial conditions reported. Training data are perturbed with Gaussian noise with standard deviation $\sigma_{noise} = k\,\sigma_{attractor}$. LSTM results for different noise levels k are plotted. **(a)** Most energetic EOF. **(b)** Second most energetic EOF. **(c)** Third most energetic EOF. **(d)** Fourth most energetic EOF.
GPR ·····; GPR-MSM ··■··; LSTM $k = 0‰$ ——; LSTM $k = 1‰$ - - -; LSTM $k = 10‰$ -⬟-; LSTM $k = 200‰$ —◆—;

F Trajectory examples

In this section we present examples of predicted trajectories of the reduced order state and compare them with the ground-truth trajectory. Moreover, we also compare the equivalent trajectories in the original space by replacing the unmodeled PCA modes with zero and projecting back to the original space. As a reference system, we pick the Kuramoto-Sivashinsky (KS) equation.

The LSTM models for both $\nu = 1/10$ and $\nu = 1/16$ have $h = 100$ hidden units and the back-propagation horizon was set to $d = 50$. An example of a trajectory obtained starting from a known short-term history of the reduced order state is plotted in Figure 9a. Moreover, the true trajectory obtained from simulating the original system, along with the evolution of the RMSE error for $\nu = 1/10$ are plotted in Figures 9b and 9c. After projecting to the original space we get the error evolution given in 9f. This error stems not only from the prediction error associated with the LSTM model used to perform forecasts but also with the error associated with the unmodeled dynamics as we only model $r_{dim} = 20$ modes. The energy included in the unmodeled modes is higher for $\nu = 1/16$ and the system exhibits higher Lyapunov exponents, as a consequence performing forecasts is a more challenging task. This can be observed in Figure 10 where the same plots are given for $\nu = 1/16$. Note that the plotted time horizon is $T = 0.4$ for $\nu = 1/16$ compared to $T = 2$ for $\nu = 1/10$.

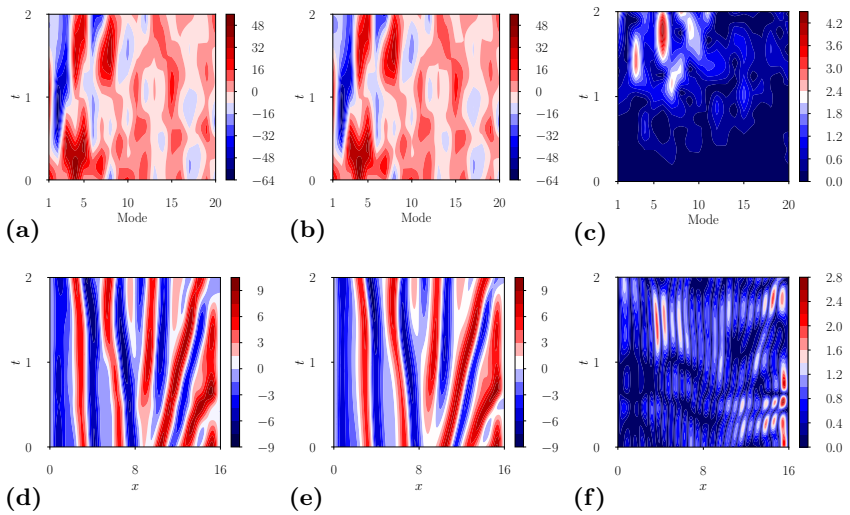

Figure 9: **(a)** Predicted evolution of the reduced order state for the KS equation with $\nu = 1/10$. **(b)** True evolution of the reduced order. **(c)** Evolution of the root mean squared error. **(d)**-**(e)** The same for the original state dimension computed by projecting to the original space replacing the unmodeled modes with zeros.

Another interesting question is how the proposed method performs when no dimensionality reduction method is used in a chaotic system with a lower Lyapunov exponent. The dynamics of this system are much easier to capture compared to the applications considered in the main paper, as the chaotic effects are less prominent and there is no missing state information. In the following, we

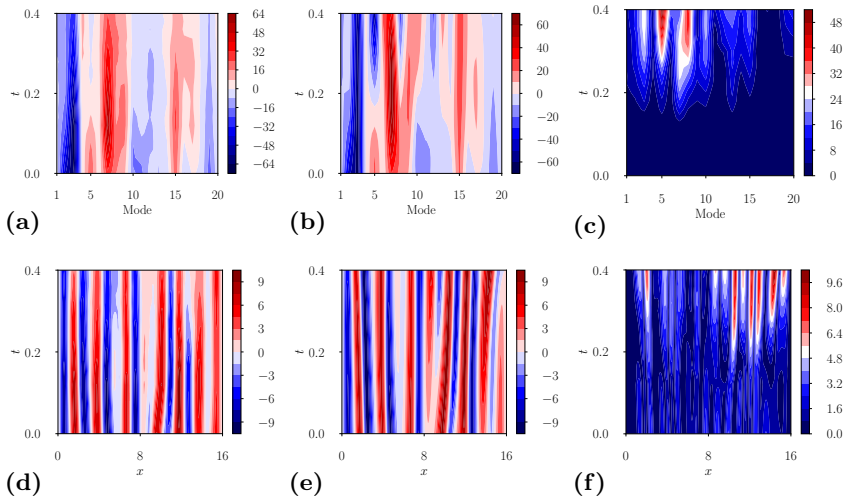

Figure 10: **(a)** Predicted evolution of the reduced order state for the KS equation with $\nu = 1/16$. **(b)** True evolution of the reduced order. **(c)** Evolution of the root mean squared error. **(d)-(e)** The same for the original state dimension computed by projecting to the original space replacing the unmodeled modes with zeros.

assume that the complete system state information is available and the LSTM forecasts the evolution of the state directly. The KS equation with $\nu = 1$ and $L = 35$ is simulated with a time-step of $dt = 0.25$ and a coarser grid with $D = 65$ points instead of $D = 513$ of the original paper. The LSTM model used has $h = 4096$ hidden units and a truncated back-propagation horizon of $d = 32$ was used. The results of three predicted trajectories along with the ground-truth and the RMSE error are illustrated in Figure 11. Note that the LSTM can forecast the evolution of the state with high accuracy for a much longer horizon compared to the results shown before where the LSTM was applied in the reduced order dimension of systems with higher Lyapunov exponents.

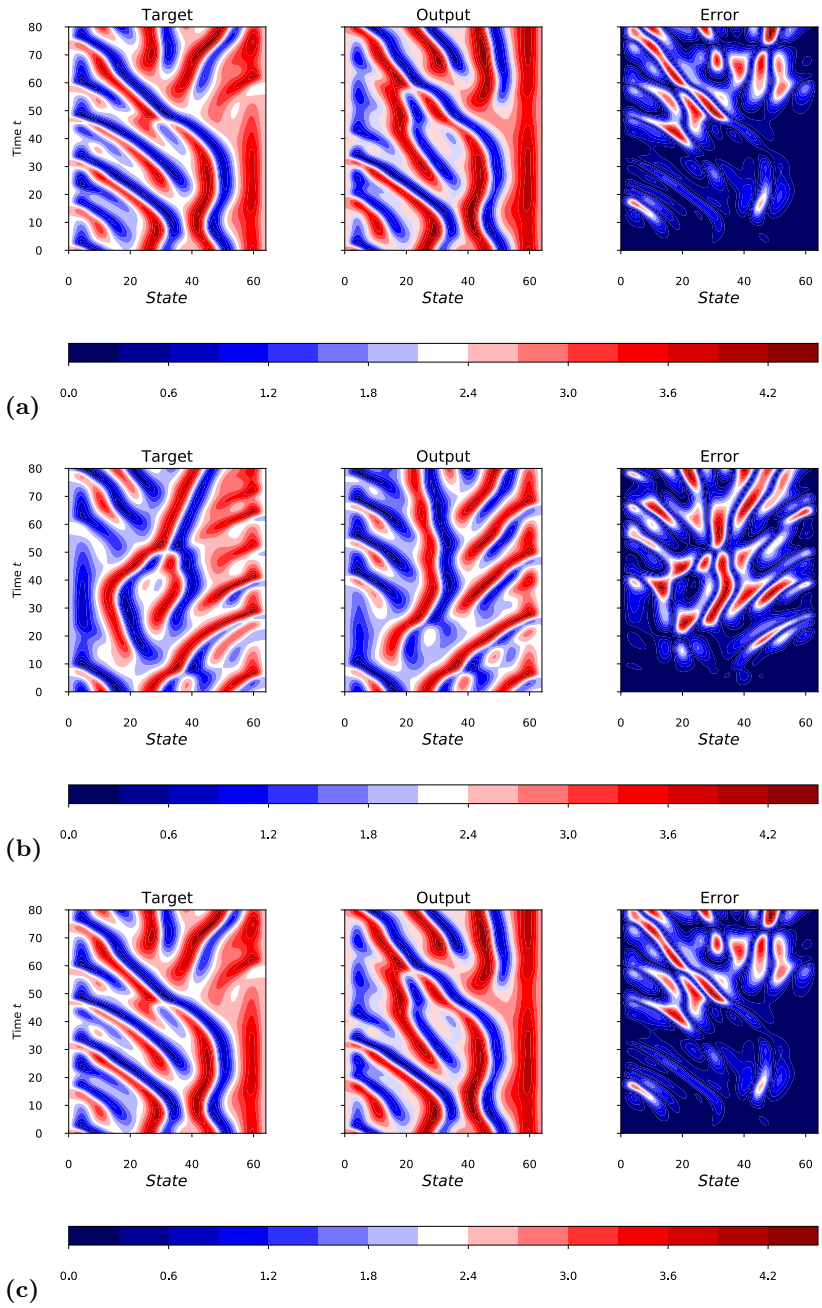

Figure 11: **(a)-(d)** True evolution of the state for the KS equation with $\nu = 1$, the predicted evolution and the associated RMSE error for three different initial conditions.