Constraining Chaos: Enforcing dynamical invariants in the training of reservoir computers

Jason A. Platt,[1] Stephen G. Penny,[2,3] Timothy A. Smith,[3,4] Tse-Chun Chen,[5] and Henry D. I. Abarbanel[a)][1,6]

[1)]*Department of Physics, University of California San Diego, San Diego, 92093, CA, USA*

[2)]*Sofar Ocean, 28 Pier Annex, San Francisco, 94105, CA, USA*

[3)]*Cooperative Institute for Research in Environmental Sciences, University of Colorado Boulder, Boulder, 80309, CO, USA*

[4)]*Physical Sciences Laboratory, National Oceanic and Atmospheric Administration, Boulder, 80305, CO, USA*

[5)]*Pacific Northwest National Laboratory, 902 Battelle Blvd, Richland, 99354, WA, USA*

[6)]*Marine Physical Laboratory, Scripps Institution of Oceanography, 9500 Gilman Drive, La Jolla, 92093, CA, USA*

(*Electronic mail: 1763platt@gmail.com)

Drawing on ergodic theory, we introduce a novel training method for machine learning based forecasting methods for chaotic dynamical systems. The training enforces dynamical invariants—such as the Lyapunov exponent spectrum and fractal dimension—in the systems of interest, enabling longer and more stable forecasts when operating with limited data. The technique is demonstrated in detail using reservoir computing, a specific kind of recurrent neural network. Results are given for the Lorenz 1996 chaotic dynamical system and a spectral quasi-geostrophic model of the atmosphere, both typical test cases for numerical weather prediction.

[a)] We express our profound sadness in the passing of Henry Don Isaac Abarbanel

Numerical weather prediction (NWP) requires vast amounts of computational power to solve global forecasting models. Machine learning (ML) systems—acting as surrogate models for the differential equations—have the promise of lessening the computational burden by using a larger time step than a classical numerical solver or through dedicated hardware acceleration. In order to be integrated into the data assimilation optimization routines of numerical weather prediction, ML models must reproduce the correct intrinsic error growth rates of the chaotic system and other dynamical invariants. We introduce here a training method for recurrent neural networks, a type of ML model that has shown great success in predicting chaotic dynamical systems, that enforces these invariants. By providing information on the physics of the system we bias the network towards reproducing the correct physical characteristics and enhance its predictive capabilities.

I. INTRODUCTION

Predicting the future trajectory of a dynamical system—a time series whose evolution is governed by a set of differential equations—is crucial in fields such as weather prediction, economics, chemistry, physics and many others[1,2]. A prediction can be generated by deriving the governing equations of motion (EOM) for the system and integrating forward in time, perhaps with data being used to determine the value of particular constants or the initial conditions. Machine learning (ML), on the other hand, allows the construction of a forecast purely from observational data in lieu of a physical model. When the EOM are expensive to evaluate numerically, ML can be used to construct a surrogate model; such models can be integrated into data assimilation[3] algorithms—such as the Kalman filter[4,5]—a typical use case when data are noisy and the model imperfect, such as in numerical weather prediction (NWP)[6].

The inclusion of physical knowledge—EOM, conservation laws and dynamical invariants—into ML algorithms has been a topic of ongoing interest[7–15]. Enforcing these laws effectively reduces the searchable parameter space for a workable model, decreasing the training time and increasing the accuracy of the resulting models. An ML model trained without knowledge of the underlying physics may fail to generalize and can produce solutions that violate fundamental constraints on the physical system[16]. Many of the examples cited above involve conservation of quantities based on the symmetry of the equations of motion, such as conservation of energy and momentum[13], or the inclusion of previously derived differential equations[11] as components of the ML training. "Physics informed" neural networks[14,15,17–19] add the known or partially known differential equations as a soft constraint in the loss function of the neural network, but conservation laws are not necessarily enforced and the equations must be known.

Many physical dynamical systems of interest are dissipative—*e.g.*, any dynamical system containing friction—meaning they exchange energy and mass with the surrounding environment[20]. High dimensional dissipative systems are very likely to exhibit chaos[21]—making them extremely sensitive to initial conditions. Enforcing conservation of quantities such as momentum, mass, or energy[11,13,22,23] for dissipative systems in isolation may not be sufficient for generalization due to the exchange of energy/momentum with the environment. Problems exhibiting chaotic dynamics, such as weather forecasting, have fractal phase space trajectories that make it difficult to specify analytic constraints[11].

With the goal of enforcing dynamical invariants, we suggest an alternative cost function for dissipative systems based on ergodicity, rather than symmetry. This has broad implications for time series prediction of dynamical systems. After formulating the invariants, we give a recurrent neural network (RNN)[24] example applied to the Lorenz 1996 system[25] and quasi-geostrophic dynamics[26] where we add soft constraints into the training of the network in order to ensure that these dynamical invariant quantities are reproduced.

II. DERIVING DYNAMICAL INVARIANTS

Ergodicity is a property of the evolution of a dynamical system. A system exhibiting ergodicity, called ergodic, is one in which the trajectories of that system will eventually visit the entire available phase space[27], with time spent in each part proportional to its volume. In general, the available phase space is a subset of the entire phase space volume. For instance a Hamiltonian system will only visit the hypersurface with constant energy[20]. Ergodicity implies that time averages over the system trajectories can be replaced with spatial averages

$$\lim_{t_f\to\infty}\frac{1}{t_f}\int_0^{t_f} g(F^t(\mathbf{u}_0))dt = \int_{\mathbf{u}\in B} g(\mathbf{u})\rho_B(\mathbf{u})du \qquad \forall \mathbf{u}_0 \in B \tag{1}$$

for an arbitrary function g, where F^t is the application of the flow of the dynamical system over time t, and $\rho_B(\mathbf{u})$ defines an invariant density over the finite set B[27]. The invariant density gives an intuitive measure of how often a trajectory visits each part of B. The invariant density defines the invariant measure[27]

$$\mu(B \subset R) = \int_B \rho_B(u) du. \tag{2}$$

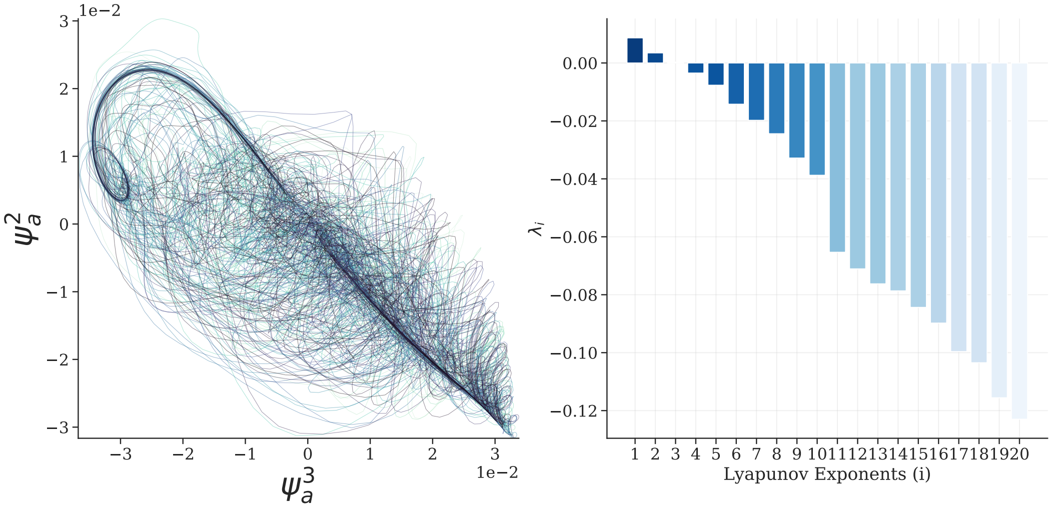

FIG. 1. left) 2D slice through the strange attractor of the spectral QG model reproduced from Fig.(3) in Reinhold and Pierrehumbert 1982[26] using the implementation in Demaeyer *et al.*2020[28]. An attractor is a hypersurface that draws in nearby trajectories of the system, such that the system will eventually be constrained to stay on the manifold. The motion of the dynamical system on a strange attractor is chaotic with extreme sensitivity to initial conditions. Strange attractors can be analyzed through the invariant measure that describes how often the system visits each part of the attractor[29,30]. Related quantities such as the fractal dimension and the Lyapunov exponents[30] are globally invariant through any smooth change of coordinates and thus are natural invariant quantities for chaotic systems. right) The Lyapunov exponents of the spectral QG model. There are two positive exponents making the system chaotic.

B will often consist of exotic geometries such as quasi-periodic orbits and strange attractors[20]. The strange attractor is a hypersurface that contains the available subspace for a chaotic dynamical system—see Fig.(1). Deterministic chaotic systems are of importance to a vast array of applications such as in NWP[3], chemical mixing[31,32], optics[33], robotics[34] and many other fields.

Despite being deterministic, the precise long term prediction of chaotic systems is impossible due to the exponential growth of errors, as quantified by the system's Lyapunov spectrum. The Lyapunov spectrum, composed of a system's Lyapunov exponents (LEs), characterizes a dynamical system[29,35] by giving a quantitative measure of how a volume of phase space stretches or shrinks over time.

For the prediction of chaotic systems, we suggest that although short term predictions will inevitably diverge, long term prediction of any system must preserve the invariants of motion characterized by the invariant measure μ_B eq.(2). Furthermore, enforcing such invariants could help to generalize the training of neural networks designed to emulate dissipative chaotic systems, in much the same way that conservation of energy and momentum has for conservative systems. While any function $g(\mathbf{u})$ integrated with the invariant density is a constant—as seen in eq.(1)—by the multiplicative ergodic theorem[36] the LEs and the fractal dimension are both invariant under smooth coordinate transformations and have algorithms that make them feasible to compute from observed data[29].

In the next sections we provide a concrete example using the fractal dimension and LEs as invariants that must be enforced when training a neural network. We use an RNN based on the reservoir computer (RC) architecture[37–39]. We impose a loss function that takes into account the preservation of the LEs and fractal dimension, and detail the benefits of doing so. We stress that the concept is not limited to RC models and can in fact apply to any neural network architecture.

III. RECURRENT NEURAL NETWORKS AND RESERVOIR COMPUTING

An RNN is a network composed of nonlinear elements that are connected in such a way as to enable self excitation[40]. Therefore, given a state of the network $\mathbf{r}(t-1)$, the next state

$$\mathbf{r}(t) = F_r(\mathbf{r}(t-1), \mathbf{u}(t-1), \theta) \tag{3}$$

is a function of the input $\mathbf{u}$, the RNN equations F_r and the internal weights θ. The label over the input data $t \in \mathbb{Z}$—conveniently called time—gives the natural order and allows the analysis of the RNN as a dynamical map. $\mathbf{r}(t)$ can then be decoded by a function $W_{\text{out}}(\mathbf{r}(t)) = \hat{\mathbf{u}}$. W_{out} is typically trained so that $\hat{\mathbf{u}}$ is as close to the target output as possible[24]. In time series prediction tasks $\hat{\mathbf{u}} \sim \mathbf{u}(t)$ so that the driven system eq.(3) can become autonomous (with no external input)

$$\mathbf{r}(t) = F_r(\mathbf{r}(t-1), W_{\text{out}}(\mathbf{r}(t-1)), \theta) \tag{4}$$

and predict the future of the dynamical system.

Reservoir computing[37,38,41–44] is a simplified form of RNN for which only the large scale parameters of the network are varied with the detailed weights selected from probability distributions. For an RC with $\tanh(\cdot)$ nonlinear activation functions at the nodes, the RNN equations become[45]

$$\mathbf{r}(t) = \alpha \tanh(\mathbf{A}\mathbf{r}(t-1) + \mathbf{W}_{\text{in}}\mathbf{u}(t-1) + \sigma_b) + (1-\alpha)\mathbf{r}(t-1). \tag{5}$$

The elements of the $N \times N$ adjacency matrix $\mathbf{A}$ are fixed *i.e.*, not trained—in contrast to other RNN architectures—with only its overall properties chosen such as the size N, density ρ_A, and spectral radius ρ_{SR}. $\mathbf{W}_{\text{in}} \in \mathbb{R}^{N\times D}$ maps the input into the high dimensional reservoir space $\mathbf{u} \in \mathbb{R}^D \rightarrow \mathbf{W}_{\text{in}}\mathbf{u} \in \mathbb{R}^N$; the elements of $\mathbf{W}_{\text{in}}$ are chosen between $\left[-\sigma, \sigma\right]$. α is the leak rate and is related to the time constant of the RC[39]. σ_b is an input bias governing the fixed point of the RC and the strength of the $\tanh(\cdot)$ nonlinearity. See Lukosevicius 2012[39] for detailed explanations of the architecture and parameter choices.

Training the RC includes training the function W_{out}—often taken to be a matrix $\mathbf{W}_{\text{out}}$ and trained through linear regression—as well as finding the correct parameters N, ρ_A, ρ_{SR}, σ, σ_b. When predicting the RC must first achieve synchronization with the input data[46] during a “spinup” phase before the prediction can begin. In[6,45,47] it is shown how to train the RC network through a two step training procedure that takes into account both the one step prediction accuracy as well as the long term forecast skill.

RC has been shown to be extremely successful in time series prediction tasks. Its simple form allows the easy computation of the Jacobian with respect to the internal (hidden) states and other quantities that can help in a dynamical systems analysis of the RNN. Platt *et al.*2021[46] showed that the RC, when well trained, can reproduce dynamical invariants such as the LEs and fractal dimension. It was further shown that the reproduction of these quantities maximized the prediction time and ensured the stability of the predictions. The training procedure used in previous works did not enforce the dynamical invariants directly, rather the results relied on using long term forecasts as a proxy. Here we reformulate the RC training to explicitly account for the dynamical invariant quantities.

IV. ENFORCING INVARIANTS

The standard training of an RC is determined by the training data and the selection of the parameters governing the global properties of the RC: N, ρ_A, ρ_{SR}, σ, σ_b and a regularization coefficient β. Once these quantities are chosen and the weights instantiated, then $\mathbf{W}_{\text{out}}$ is given

$$\mathbf{W}_{\text{out}} = \mathbf{u}\mathbf{r}^T(\mathbf{r}\mathbf{r}^T + \beta\mathbb{I})^{-1}$$

where $\mathbf{u}$ is the $D \times T$ matrix of input data, T is the number of time steps and $\mathbf{r}$ is the $N \times T$ matrix of reservoir states[45]. We can add into the selection of these parameters prior knowledge of the global

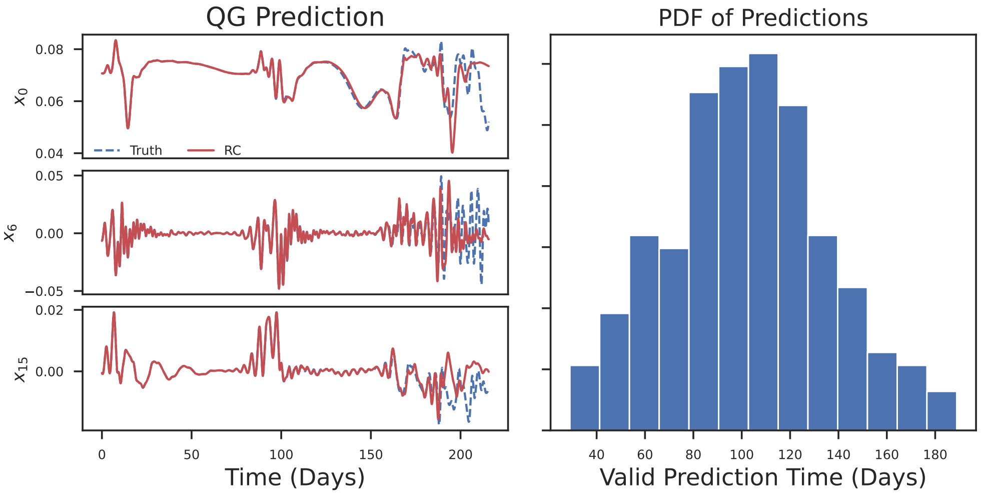

FIG. 2. Example prediction and the probability distribution function of the valid prediction time (VPT) for 200 initial conditions over the 20 dimensional QG system described in section V B. VPT[6,48] is calculated as the time when the root mean square error $\mathrm{RMSE(t)} = \sqrt{\frac{1}{\mathrm{D}}\sum_{\mathrm{i=1}}^{\mathrm{D}}\left[\frac{\mathrm{u_i^f(t)-u_i(t)}}{\sigma_\mathrm{i}}\right]^2}$ exceeds a certain value ε, in this case $\varepsilon = 0.3$ approximately in line with[6,45,48]. D is the system dimension, σ is the long term standard deviation of the time series and u^f is the RC forecast. The largest LE $1/\lambda_1 \sim 12$ days gives the natural time scale for error growth of the system and thus can be used as a measurement for the predictive skill. The RC returns fantastic prediction times for this low resolution model. The size of the RC is N=1500 and 100,000 training steps were provided with a $\Delta t = 80min$ giving about 30 years of data, we consider this the "data rich" case because extra training data does not have a significant impact on the predictive ability of the network.

invariants of the system $\mathbf{u}$ and a penalty term for longer-range forecasts. Therefore we construct a loss function

$$\mathrm{Loss} = \varepsilon_1 \|\mathrm{C_u} - \mathrm{C_{RC}}\|^2 + \varepsilon_2 \sum_{\mathrm{k=1}}^{\mathrm{M}} \sum_{\mathrm{t=t_i}}^{\mathrm{t_f}} \left\|\mathbf{u}_\mathrm{k}^\mathrm{f}(\mathrm{t}) - \mathbf{u}_\mathrm{k}(\mathrm{t})\right\|^2 \exp\left\{-\frac{\mathrm{t-t_i}}{\mathrm{t_f-t_i}}\right\}; \mathrm{t} \in \mathbb{Z} \tag{6}$$

that can be minimized over the RC parameters and with ε_x hyperparameters. The selection of the parameters leads to the matrix $\mathbf{W}_{\mathrm{out}}$ based on training data $\mathbf{u}_{\mathrm{train}}$. Platt *et al.*2022[45] generated a number of long term forecasts $\mathbf{u}^f(t)$ and compared them to the data $\mathbf{u}$; with enough data this procedure often leads to a model that reproduces the correct dynamical invariants. Without the explicit enforcement of these invariants, however, the model can fail to capture the dynamics—particularly for high dimensional systems and in cases where the number of trajectories M is constrained. Here we add the dynamical invariants (C_x) as a constraint in order to directly train for generalizability, similar to Beucler *et al.*2021[49]. This scheme is illustrated in Fig.(3). The global optimization routine used to minimize the cost function was the covariance matrix adaption evolution strategy (CMA-ES)[50,51]. CMA-ES is a derivative free global optimization routine involving the iterative update of sampled points through an evolutionary strategy. At each iteration only the sampled points with lowest loss are kept and that distribution is used to update the mean and covariance matrix of the routine from which new samples are drawn. As we are optimizing a non-convex function, the technique is not guaranteed to converge to the global optimum in the given time specified for optimization—indeed we often see differences in the parameters found by the routine depending on the initial sampling.

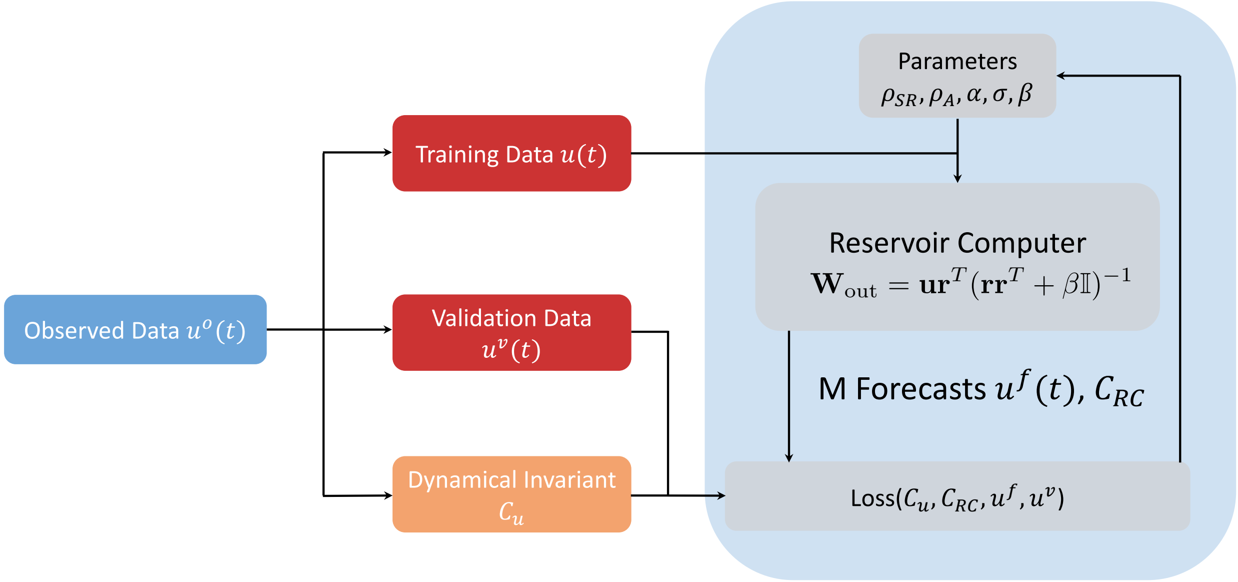
Parameters
$\rho_{SR}, \rho_A, \alpha, \sigma, \beta$
Training Data $u(t)$
Reservoir Computer
$\mathbf{W}_{\text{out}} = \mathbf{u}\mathbf{r}^T(\mathbf{r}\mathbf{r}^T + \beta\mathbb{I})^{-1}$
Observed Data $u^o(t)$
Validation Data $u^v(t)$
M Forecasts $u^f(t)$, C_{RC}
Dynamical Invariant C_u
Loss(C_u, C_{RC}, u^f, u^v)

FIG. 3. Parameter optimization of a reservoir computer showing the introduction of dynamical invariants. The observed data is split into training, validation and testing sets with the invariants calculated from the data[29]. These quantities can then be incorporated into the loss function to improve the overall training of the RC. A general discussion of the training strategy is found in Platt *et al.*2022[45].

We use the Lyapunov exponents and the fractal dimension as examples of dynamical invariants in order to demonstrate the technique. With the equations of motion, such as eq.(4) for the RC, it is quite simple to calculate these quantities using well known and efficient algorithms[30]. When training directly from data—without knowledge of the underlying system—we may not know the equations of motion so these quantities must be estimated. The largest LE can often be approximated from time series data[29,52,53] and the fractal dimension can be calculated using various techniques[29,54]. A calculation of the full LE spectrum is more difficult. Use of other dynamical invariants derived from the invariant measure eq.(2) are also possible, for instance the energy density spectrum of a fluid dynamical system as a function of wavenumber.

V. RESULTS

A. Lorenz 1996

Our first test case for the RC is the Lorenz 1996 system (L96)[25], a standard testbed for data assimilation applications in numerical weather prediction. L96 describes the evolution of a scalar quantity over a number of sites positioned uniformly over a periodic lattice of constant latitude. The evolution of this scalar quantity is governed by terms representing advection and diffusion,

$$\frac{\mathrm{d}u_k}{\mathrm{d}t} = -u_{k-1}(u_{k-2} - u_{k+1}) - u_k + F. \tag{7}$$

In this case we take the number of sites to be $D = 10$ and forcing $F = 8.0$ with the purpose of making the system hyperchaotic, with three positive Lyapunov exponents Fig.(4).

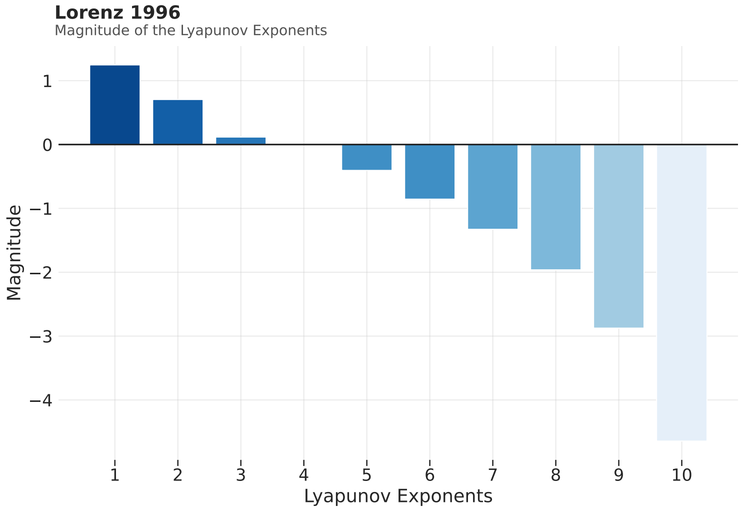

FIG. 4. Lyapunov exponents of the 10D Lorenz 1996 system[25] with $F = 8.0$. There are three positive LEs, a single zero exponent, and the remaining LEs are negative.

The results for $C_{RC} = LEs$ eq.(6) are shown in Fig.(5). When no global information is given to the RC then it can fail to generalize when presented with unseen input. Simply providing the largest LE to the RC during training enables the neural networks to:

1. generalize to unseen data so that there are good predictions over the entire range of possible initial conditions, and

2. reconstruct the attractor as in[55] and[46] with the prediction giving the correct ergodic properties of the data even after the prediction necessarily diverges from the ground truth.

In this case, providing the largest LE was sufficient to improve the predictions, with no further gains coming from providing additional LEs. Indeed, a slight decrease in the mean prediction time is observed when providing more LEs. A calculation of the LEs of the reservoir after optimization shows that providing the single largest exponent and the sample trajectories are sufficient for the entire LE spectrum to be reproduced. Providing additional LEs during training therefore adds more constraints but no new information, resulting in a slightly less optimal result.

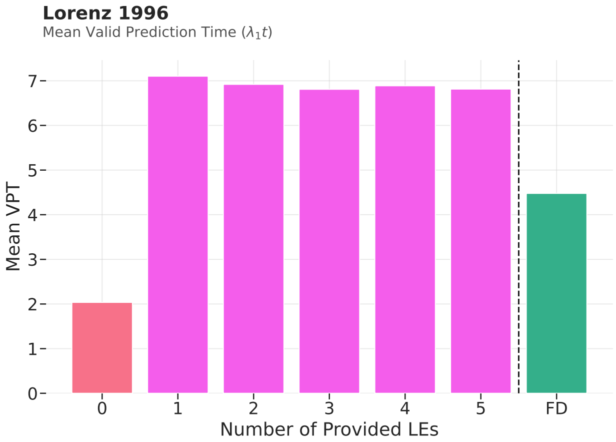

FIG. 5. The RC is initialized 10 times, providing k LEs and M=7 long term forecasts eq.(6); we report on the average of the distribution of predictions for the 10 dimensional Lorenz 1996 system[25] eq.(7). There are 10 total LEs—3 positive, shown in Fig.(4). When 0 LEs are provided the RC has no global information and the prediction time is poor. Providing the largest LE is sufficient for generalizable predictions and this quantity is quite easily calculated from numerical data[29]. The last column shows the alternative result when providing only the fractal dimension (no LEs provided) as another example of an invariant quantity. The size of the RC is $N = 400$ and the number of test initial conditions is 1000. The forecast time is given as an average over the 1000 trajectories and then scaled by the largest LE to give the prediction in terms of the number of Lyapunov timescales. The LEs were calculated using the QR decomposition method[56] from the equations of motion. Results of how the forecast and hyperparameter distributions change as the number of exponents is varied are found in the supplemental information.

The second invariant given is the fractal dimension of the data calculated through the Kaplan-Yorke formulation[57,58]

$$\text{Dimension} = \alpha + \frac{\sum_{i=1}^{\alpha} \lambda_i}{|\lambda_{\alpha+1}|}, \tag{8}$$

with D the system dimension, λ the ordered LEs and α the smallest index where the sum of the LEs does not cross zero[29]. There are alternate definitions and methods for calculating the fractal dimension[54].

Providing the fractal dimension to the loss function as an invariant has a similar effect to providing the LEs by raising the mean valid prediction time (VPT) from $\sim 2 \rightarrow 4$. The fractal dimension may not, however, be unique to a particular set of data while the LE spectrum has a much greater chance of constraining the shape of the resulting strange attractor. Therefore, we see that there is not as much improvement in the forecast when providing the fractal dimension compared to the LE spectrum.

It is typically possible to calculate an approximation of the largest LE from numerical data[29]—for a large and complex system, however, the calculation is often difficult and the result may be inaccurate. The robustness of the proposed training technique to errors in provided LEs is tested in Fig(6), where multiples of the correct exponent are provided to the RC during training. The addition of the LE acts as a guide to the training, not as a hard requirement—see eq.(6). Errors in the provided exponent up to a factor of 2-3 cause only a slight degradation of performance. This experiment shows that the training method is robust to the numerical errors in the calculation of the largest LE from measured data.

B. Synoptic Scale Atmospheric Model

For more complex higher-dimensional dynamical systems, the Lyapunov spectrum or Kaplan-Yorke dimension are quite difficult, if not impossible, to calculate. However, our previous results showed that capturing the leading Lyapunov exponent (LLE) enhanced prediction skill greatly, and even with complex models this quantity can be estimated more readily either from data[52,53] or from a model[56,59]. We therefore explore the value of representing the LLE in the case of quasi-geostrophic (QG) dynamics[60], which assume that large-scale atmospheric disturbances are governed by the conservation of potential temperature and absolute potential vorticity, while the horizontal velocity is quasi-geostrophic and the pressure quasi-hydrostatic. Numerical models

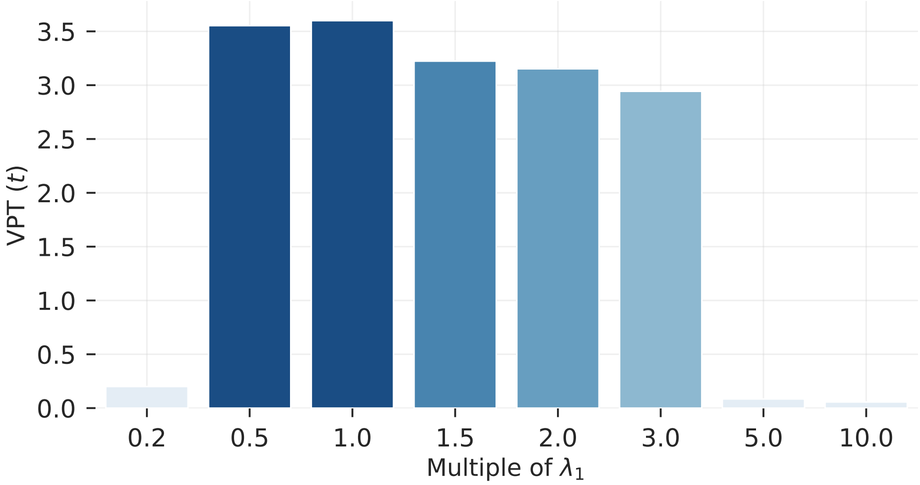

FIG. 6. The effect of including an incorrect LE in the training for the 10D Lorenz 96 system. The RC dimension is N=400 with 1 LE and M=7 trajectories with $t_f = 3$ provided during training eq.(6). The peak prediction time occurs when the exact exponent is given but even large errors of up to a factor of 2-3 cause only minimal degradation to the prediction skill.

based on the QG approximation were a precursor to larger scale primitive equation models used for global numerical weather prediction[3] and are frequently used in data assimilation studies targeting the atmosphere and ocean[61,62]. Here, we consider the two-layer baroclinic model of Charney and Strauss (1980)[63] used to study the planetary-scale motions of a thermally driven atmosphere in the presence of topography. We further incorporate the adaption of Reinhold and Pierrehumbert (1982)[26] to include an additional wave in the zonal direction making it highly baroclinically unstable. We use the implementation of Demaeyer *et al.*(2020)[28], which provides a truncated 2-layer QG atmospheric model on a mid-latitude β-plane frictionally coupled to a mountain and a valley with a dimension of 20 in the spectral space of the model.

For the atmospheric streamfunctions ψ_a^1/ψ_a^3 at heights 250/750 hPa and the vertical velocity $\omega = \frac{\mathrm{d}p}{\mathrm{d}t}$, the equations of motion are derived to be

$$\frac{\partial}{\partial t}\overbrace{\left(\nabla^2\psi_a^1\right)}^{\text{vorticity}} + \overbrace{J(\psi_a^1,\nabla^2\psi_a^1)}^{\text{horizontal advection}} + \overbrace{\beta\frac{\partial\psi_a^1}{\partial x}}^{\beta-\text{plane Coriolis force}} = -\overbrace{k_d'\nabla^2(\psi_a^1-\psi_a^3)}^{\text{friction}} + \overbrace{\frac{f_0}{\Delta p}\omega}^{\text{vertical stretching}} \tag{9}$$

$$\frac{\partial}{\partial t}\left(\nabla^2\psi_a^3\right) + J(\psi_a^3,\nabla^2\psi_a^3) + J(\psi_a^3, f_0 h/H_a) + \beta\frac{\partial\psi_a^3}{\partial x} = k_d'\nabla^2(\psi_a^1-\psi_a^3) - k_d\nabla^2\psi_a^3 + \frac{f_0}{\Delta p}\omega \tag{10}$$

with $\nabla = \frac{\partial}{\partial x}\hat{x} + \frac{\partial}{\partial y}\hat{y}$, k_d' the friction between the layers, k_d the friction between the atmosphere and the ground, h/H_a the ratio of ground height to the characteristic depth of the atmospheric layer, $\Delta p = 500$ hPa the pressure differential between the layers and $J(g_1, g_2) = \frac{\partial g_1}{\partial x}\frac{\partial g_2}{\partial y} - \frac{\partial g_1}{\partial y}\frac{\partial g_2}{\partial x}$ the Jacobian. More details are given in[26,28].

After integrating the model forward in time, we ask if an RC model is capable of predicting the dynamics given a significant amount of training data. An example forecast and distribution of predictions is shown in Fig.(2) where the conventionally trained[6,45] RC model successfully predicts the synoptic scale atmospheric dynamics for a number of months. Such significant predictive power on a low resolution QG model is an interesting result in and of itself, showcasing the ability of RNNs to resolve more realistic atmospheric dynamics.

When we reduce the amount of data and set $M = 1$—the limited data case in eq. (6) with a single forecast as part of the training loss—the RC model loses its predictive power. Adding in the information contained in the LLE, which is ~ 0.01, enables the model to recover a large amount of predictive capability. The calculated LLE of the RC with no provided LEs is 0.14, compared to ~ 0.01 when it is provided. The mismatch between the LEs of the two systems is a clear indication that synchronization between the two is not achieved[46]. The impact of the added information provided via the LLE is clear in Figure 7, where the average VPT has extended from only a few days to multiple weeks.

VI. DISCUSSION AND CONCLUSION

Chaotic dynamical systems are difficult to predict due to their sensitivity to initial conditions[64]. Better understanding and accounting for dynamical uncertainties has, however, allowed fields like numerical weather prediction to provide useful and continually improving forecasts[65]. Previous works (e.g.[13,49]) proposed that including conserved quantities such as energy/momentum may help to improve the application of neural networks to physical systems. However, the introduction of the proposed conserved quantities is not generally applicable to dissipative chaotic dynamical systems. Thus, we instead considered dynamical invariants based on the invariant measure.

We provided a concrete example using quantities derived from the invariant measure, such as the Lyapunov exponents and the fractal dimension, to train a particular RNN architecture called reservoir computing (RC). Previous RC training algorithms used long-term forecasts initialized from many different initial conditions in order to improve generalizability[6,47], essentially imposing these invariant measures by proxy. Here, we imposed the invariant measures as constraints

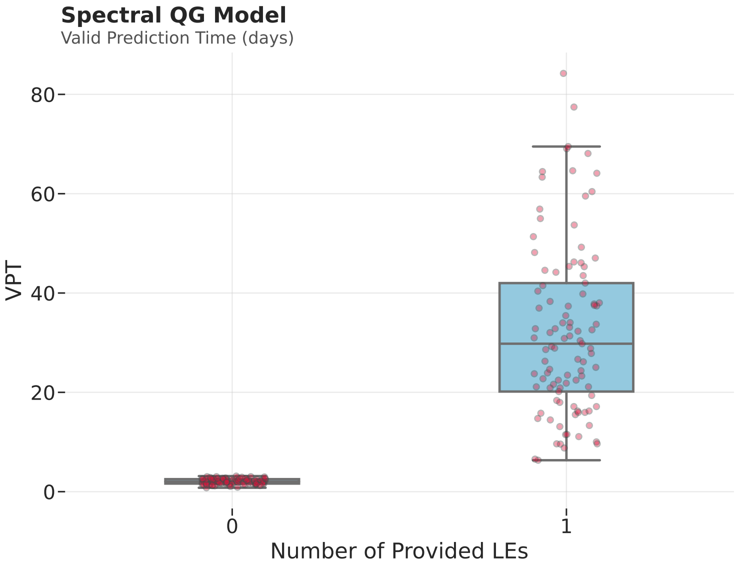

FIG. 7. Prediction time in days across 100 predictions for the QG model with an RC trained with and without input of LEs and with $t_f = 1$ in the optimization routine eq.(6). The size of the reservoir is $N = 500$ and 50000 training steps are provided with $\Delta t = 80min$. Without the provided exponents in this low information regime, unlike in the data rich example in Fig(2), the RC model is unable to correctly infer the correct dynamical invariants and fails to generalize.

directly in the training algorithm, allowing the RC to generalize with fewer data. Fortunately, we have found that much of the value of this additional constraint is achieved through the use of the leading Lyapunov exponent. While the entire Lyapunov spectrum can be quite difficult to calculate, particularly for large systems, the leading Lyapunov exponent can be estimated by using numerical techniques such as the breeding method[66,67] or other methods described in[21,56,59]. This provides an opportunity for extension of this technique to higher-dimensional systems.

Recent works from[68–70] have shown promise in producing data-driven surrogate weather models that are competitive by some metrics with conventional operational forecast models. A key property that has not yet been demonstrated with such surrogate models is their ability to reproduce dynamical quantities such as the LEs, which indicate an average measure of the response to small errors in the initial conditions. For weather models in particular, the enforcement of LEs is crucial for the correct operation of data assimilation algorithms[3]. Platt *et al.*2021[46] demonstrated the importance of reconstructing the LE spectrum for producing a skillful deterministic forecast model. Similarly, Penny *et al.*[6] indicated the ability of the RC to reproduce accurate finite-time LEs as a requirement for RC-based ensemble forecasts to produce good estimates of the forecast error covariance, which is the primary tool used in conventional data assimilation methods to project observational data to unobserved components of the system. This information can then be used to make the RC robust to sparse and noisy measurements. While this is the more realistic scenario used in online weather prediction systems, it is a fact that is rarely taken into account in neural network applications. The introduction of explicit constraints in the training cost function both improves predictions and trains the RNN to reconstruct the correctly shaped attractor[45,55].

VII. ACKNOWLEDGEMENTS

J.A. Platt, S.G. Penny, and H.D.I. Abarbanel acknowledge support from the Office of Naval Research (ONR) grants N00014-19-1-2522 and N00014-20-1-2580. S.G. Penny and T.A. Smith acknowledge support from NOAA grant NA20OAR4600277. T.-C. Chen acknowledges support from the NOAA Cooperative Agreement with the Cooperative Institute for Research in Environmental Sciences at the University of Colorado Boulder, NA17OAR4320101.

VIII. SOURCE CODE

The basic RC implementation used in this study is available
https://github.com/japlatt/BasicReservoirComputing

REFERENCES

[1] H. Abarbanel, Predicting the future: completing models of complex systems, Springer, New York, 2013.

[2] S. H. Strogatz, Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry and Engineering, Westview Press, 2000.

[3] E. Kalnay, Atmospheric Modeling, Data Assimilation and Predictability, Cambridge University Press, 2002. doi:10.1017/CBO9780511802270.

[4] R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems, Journal of Basic Engineering 82 (1) (1960) 35–45. doi:10.1115/1.3662552.

[5] J. Mandel, A brief tutorial on the ensemble kalman filter, arXiv:0901.3725 (2009).

[6] S. G. Penny, T. A. Smith, T.-C. Chen, J. A. Platt, H.-Y. Lin, M. Goodliff, H. D. I. Abarbanel, Integrating recurrent neural networks with data assimilation for scalable data-driven state estimation, Journal of Advances in Modelling Earth Systems (2022). doi:10.1029/2021MS002843.

[7] G. Karniadakis, Y. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning (2021) 1–19doi:10.1038/s42254-021-00314-5.

[8] S. Ha, H. Jeong, Discovering conservation laws from trajectories via machine learning (2021). doi:10.48550/ARXIV.2102.04008.
URL https://arxiv.org/abs/2102.04008

[9] F. Alet, D. D. Doblar, A. Zhou, J. B. Tenenbaum, K. Kawaguchi, C. Finn, Noether networks: Meta-learning useful conserved quantities, ArXiv abs/2112.03321 (2021).

[10] Z. Liu, M. Tegmark, Machine learning conservation laws from trajectories, Phys. Rev. Lett. 126 (2021) 180604. doi:10.1103/PhysRevLett.126.180604.
URL https://link.aps.org/doi/10.1103/PhysRevLett.126.180604

[11] T. Beucler, M. Pritchard, S. Rasp, J. Ott, P. Baldi, P. Gentine, Enforcing analytic constraints in neural networks emulating physical systems, Phys. Rev. Lett. 126 (2021) 098302. doi:10.1103/PhysRevLett.126.098302.
URL https://link.aps.org/doi/10.1103/PhysRevLett.126.098302

[12] Z. Chen, J. Zhang, M. Arjovsky, L. Bottou, Symplectic recurrent neural networks, in: International Conference on Learning Representations, 2020.
URL https://openreview.net/forum?id=BkgYPREtPr

[13] S. Greydanus, M. Dzamba, J. Yosinski, Hamiltonian neural networks, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019.
URL https://proceedings.neurips.cc/paper/2019/file/26cd8ecadce0d4efd6cc8a8725cbd1f8-Paper.pdf

[14] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707. doi:https://doi.org/10.1016/j.jcp.2018.10.045.
URL https://www.sciencedirect.com/science/article/pii/S0021999118307125

[15] L. Yang, X. Meng, G. E. Karniadakis, B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data, Journal of Computational Physics 425 (2021)

109913. doi:https://doi.org/10.1016/j.jcp.2020.109913.

[16] A. Azizi, M. Pleimling, A cautionary tale for machine learning generated configurations in presence of a conserved quantity, Scientific Reports 11 (1) (2021) 6395. doi:10.1038/s41598-021-85683-8.

[17] N. Doan, W. Polifke, L. Magri, Physics-informed echo state networks, Journal of Computational Science 47 (2020) 101237. doi:https://doi.org/10.1016/j.jocs.2020.101237.

[18] N. A. K. Doan, W. Polifke, L. Magri, Short- and long-term predictions of chaotic flows and extreme events: a physics-constrained reservoir computing approach, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 477 (2253) (2021) 20210135. doi:10.1098/rspa.2021.0135.

[19] A. Racca, L. Magri, Automatic-differentiated physics-informed echo state network (api-esn), in: M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, P. M. Sloot (Eds.), Computational Science – ICCS 2021, Springer International Publishing, Cham, 2021, pp. 323–329.

[20] H. Goldstein, C. Poole, J. Safko, Classical Mechanics, 3rd Edition, Addison Wesley, 2001.

[21] I. Ispolatov, V. Madhok, S. Allende, M. Doebeli, Chaos in high-dimensional dissipative dynamical systems, Scientific Reports (2015). doi:10.1038/srep12506.

[22] L. Zanna, T. Bolton, Data-Driven Equation Discovery of Ocean Mesoscale Closures, Geophysical Research Letters 47 (17) (2020) e2020GL088376, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2020GL088376. doi:10.1029/2020GL088376.
URL http://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020GL088376

[23] T. Beucler, M. Pritchard, P. Gentine, S. Rasp, Towards Physically-Consistent, Data-Driven Models of Convection, in: IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 2020, pp. 3987–3990, iSSN: 2153-7003. doi:10.1109/IGARSS39084.2020.9324569.

[24] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.

[25] E. N. Lorenz, Predictability: A problem partly solved, in: T. Palmer, R. Hagedorn (Eds.), Predictability of weather and climate, Cambridge, 2006.

[26] B. B. Reinhold, R. T. Pierrehumbert, Dynamics of weather regimes: Quasi-stationary waves and blocking, Monthly Weather Review 110 (9) (1982) 1105 – 1145. doi:10.1175/

1520-0493(1982)110<1105:DOWRQS>2.0.CO;2.

[27] G. Datseris, U. Parlitz, Nonlinear Dynamics: A Concise Introduction Interlaced with Code, 1st Edition, Springer, 2022.

[28] J. Demaeyer, L. De Cruz, S. Vannitsem, qgs: A flexible python framework of reduced-order multiscale climate models, Journal of Open Source Software (2020). doi:10.21105/joss.02597.

[29] H. D. I. Abarbanel, The Analysis of Observed Chaotic Data, Springer-Verlag, New York, 1996.

[30] J. P. Eckmann, D. Ruelle, Ergodic theory of chaos and strange attractors, Rev. Mod. Phys. 57 (1985) 617–656. doi:10.1103/RevModPhys.57.617.
URL https://link.aps.org/doi/10.1103/RevModPhys.57.617

[31] J. M. Ottino, S. R. Wiggins, M. R. Bringer, C. J. Gerdts, H. Song, J. D. Tice, R. F. Ismagilov, Microfluidic systems for chemical kinetics that rely on chaotic mixing in droplets, Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 362 (1818) (2004) 1087–1104. doi:10.1098/rsta.2003.1364.

[32] R. O. Grigoriev, M. F. Schatz, V. Sharma, Chaotic mixing in microdroplets, Lab Chip 6 (2006) 1369–1372. doi:10.1039/B607003E.

[33] F. T. Arecchi, G. Giacomelli, P. L. Ramazza, S. Residori, Experimental evidence of chaotic itinerancy and spatiotemporal chaos in optics, Phys. Rev. Lett. 65 (1990) 2531–2534. doi:10.1103/PhysRevLett.65.2531.
URL https://link.aps.org/doi/10.1103/PhysRevLett.65.2531

[34] X. Zang, S. Iqbal, Y. Zhu, X. Liu, J. Zhao, Applications of chaotic dynamics in robotics, International Journal of Advanced Robotic Systems 13 (2) (2016) 60. doi:10.5772/62796.

[35] A. M. Lyapunov, The general problem of the stability of motion, International Journal of Control 55 (3) (1992) 531–534. doi:10.1080/00207179208934253.

[36] V. I. Oseledec, A multiplicative ergodic theorem. lyapunov characteristic numbers for dynamical systems, Trudy Mosk. Mat. Obsc. 19 (1968) 197–.

[37] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Computation 14 (2002) 2531–2560.

[38] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks-with an erratum note, Bonn, Germany: German National Research Center for Information Technology GMD Technical Report 148 (2010) 1–47.

[39] M. Lukosevicius, A practical guide to applying echo state networks (2012) 659–686doi:10.1007/978-3-642-35289-8_36.

[40] J. L. Elman, Finding structure in time, Cognitive Science 14 (2) (1990) 179–211. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1207/s15516709cog1402_1, doi:https://doi.org/10.1207/s15516709cog1402_1.
URL https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1

[41] H. Jaeger, Short term memory in echo state networks. gmd-report 152, in: GMD - German National Research Institute for Computer Science, 2002.

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

[43] M. Lukosevicius, H. Jaeger, Reservoir computing approaches to recurrent neural network training, Comput. Sci. Rev. 3 (2009) 127–149.

[44] H. Jaeger, Long short-term memory in echo state networks: Details of a simulation study (2012).

[45] J. A. Platt, S. G. Penny, T. A. Smith, T.-C. Chen, H. D. I. Abarbanel, A systematic exploration of reservoir computing for forecasting complex spatiotemporal dynamics (2022). arXiv:2201.08910.

[46] J. A. Platt, A. S. Wong, R. Clark, S. G. Penny, H. D. I. Abarbanel, Robust forecasting using predictive generalized synchronization in reservoir computing, Chaos 31 (2021) 123118.
URL https://doi.org/10.1063/5.0066013

[47] A. Griffith, A. Pomerance, D. J. Gauthier, Forecasting chaotic systems with very low connectivity reservoir computers, Chaos: An Interdisciplinary Journal of Nonlinear Science 29 (12) (2019) 123108. doi:10.1063/1.5120710.

[48] P. Vlachas, J. Pathak, B. Hunt, T. Sapsis, M. Girvan, E. Ott, P. Koumoutsakos, Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics, Neural Networks 126 (2020) 191 – 217. doi:https://doi.org/10.1016/j.neunet.2020.02.016.

[49] T. Beucler, M. Pritchard, S. Rasp, J. Ott, P. Baldi, P. Gentine, Enforcing Analytic Constraints in Neural Networks Emulating Physical Systems, Physical Review Letters 126 (9) (2021) 098302, publisher: American Physical Society. doi:10.1103/PhysRevLett.126.098302.
URL https://link.aps.org/doi/10.1103/PhysRevLett.126.098302

[50] N. Hansen, S. D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es), Evolutionary Computation 11 (1)

(2003) 1–18. doi:10.1162/106365603321828970.

[51] N. Hansen, The cma evolution strategy: A tutorial (2023). arXiv:1604.00772.

[52] M. T. Rosenstein, J. J. Collins, C. J. De Luca, A practical method for calculating largest lyapunov exponents from small data sets, Physica D: Nonlinear Phenomena 65 (1) (1993) 117–134. doi:https://doi.org/10.1016/0167-2789(93)90009-P.

[53] H. Kantz, A robust method to estimate the maximal lyapunov exponent of a time series, Physics Letters A 185 (1) (1994) 77–87. doi:https://doi.org/10.1016/0375-9601(94)90991-1.

[54] J. Theiler, Estimating fractal dimension, Journal of The Optical Society of America A-optics Image Science and Vision 7 (1990) 1055–1073.

[55] Z. Lu, B. R. Hunt, E. Ott, Attractor reconstruction by machine learning, Chaos: An Interdisciplinary Journal of Nonlinear Science 28 (6) (2018) 061104. doi:10.1063/1.5039508.

[56] K. Geist, U. Parlitz, W. Lauterborn, Comparison of Different Methods for Computing Lyapunov Exponents, Progress of Theoretical Physics 83 (5) (1990) 875–893. doi:10.1143/PTP.83.875.

[57] J. L. Kaplan, J. A. Yorke, Chaotic behavior of multidimensional difference equations, in: H.-O. Peitgen, H.-O. Walther (Eds.), Functional Differential Equations and Approximation of Fixed Points, Springer Berlin Heidelberg, Berlin, Heidelberg, 1979, pp. 204–227.

[58] P. Frederickson, J. L. Kaplan, E. D. Yorke, J. A. Yorke, The liapunov dimension of strange attractors, Journal of Differential Equations 49 (2) (1983) 185 – 207. doi:https://doi.org/10.1016/0022-0396(83)90011-6.

[59] G. Benettin, L. Galgani, J.-M. Strelcyn, Kolmogorov entropy and numerical experiments, Phys. Rev. A 14 (1976) 2338–2345. doi:10.1103/PhysRevA.14.2338.
URL https://link.aps.org/doi/10.1103/PhysRevA.14.2338

[60] J. G. Charney, On the scale of atmospheric motions, Geofysiske Publikasjoner 17 (1948).

[61] S.-C. Yang, M. Corazza, A. Carrassi, E. Kalnay, T. Miyoshi, Comparison of local ensemble transform kalman filter, 3dvar, and 4dvar in a quasigeostrophic model, Monthly Weather Review 137 (2) (2009) 693 – 709. doi:https://doi.org/10.1175/2008MWR2396.1.

[62] K. Swanson, R. Vautard, C. Pires, Four-dimensional variational assimilation and predictability in a quasi-geostrophic model, Tellus A: Dynamic Meteorology and Oceanography 50 (4) (1998) 369–390. doi:10.3402/tellusa.v50i4.14540.

[63] J. G. Charney, D. M. Straus, Form-drag instability, multiple equilibria and propagating planetary waves in baroclinic, orographically forced, planetary wave systems, Journal of Atmospheric

Sciences 37 (6) (1980) 1157 – 1176. doi:https://doi.org/10.1175/1520-0469(1980)037<1157:FDIMEA>2.0.CO;2.

[64]E. N. Lorenz, Deterministic Nonperiodic Flow, Journal of the Atmospheric Sciences 20 (2) (1963) 130–141. doi:10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

[65]E. Kalnay, Atmospheric modeling, data assimilation, and predictability, Cambridge University Press, 2003.

[66]Z. Toth, E. Kalnay, Ensemble forecasting at nmc: The generation of perturbations, Bulletin of the American Meteorological Society 74 (12) (1993) 2317 – 2330. doi:10.1175/1520-0477(1993)074<2317:EFANTG>2.0.CO;2.
URL https://journals.ametsoc.org/view/journals/bams/74/12/1520-0477_1993_074_2317_efantg_2_0_co_2.xml

[67]Z. Toth, E. Kalnay, Ensemble forecasting at ncep and the breeding method, Monthly Weather Review 125 (12) (1997) 3297 – 3319. doi:10.1175/1520-0493(1997)125<3297:EFANAT>2.0.CO;2.
URL https://journals.ametsoc.org/view/journals/mwre/125/12/1520-0493_1997_125_3297_efanat_2.0.co_2.xml

[68]J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, P. Hassanzadeh, K. Kashinath, A. Anandkumar, Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators (2022). arXiv:2202.11214.

[69]R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, A. Pritzel, S. Ravuri, T. Ewalds, F. Alet, Z. Eaton-Rosen, W. Hu, A. Merose, S. Hoyer, G. Holland, J. Stott, O. Vinyals, S. Mohamed, P. Battaglia, Graphcast: Learning skillful medium-range global weather forecasting (2022). arXiv:2212.12794.

[70]K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, Q. Tian, Pangu-weather: A 3d high-resolution model for fast and accurate global weather forecast (2022). arXiv:2211.02556.

Appendix A: Supplemental Information

We provide in this supplemental section a few detailed graphs that give added insight into the working of the optimization routine. Figure (8) more clearly shows how the additional information in the dynamical invariants enables the routine to find the correct set of parameters. Even

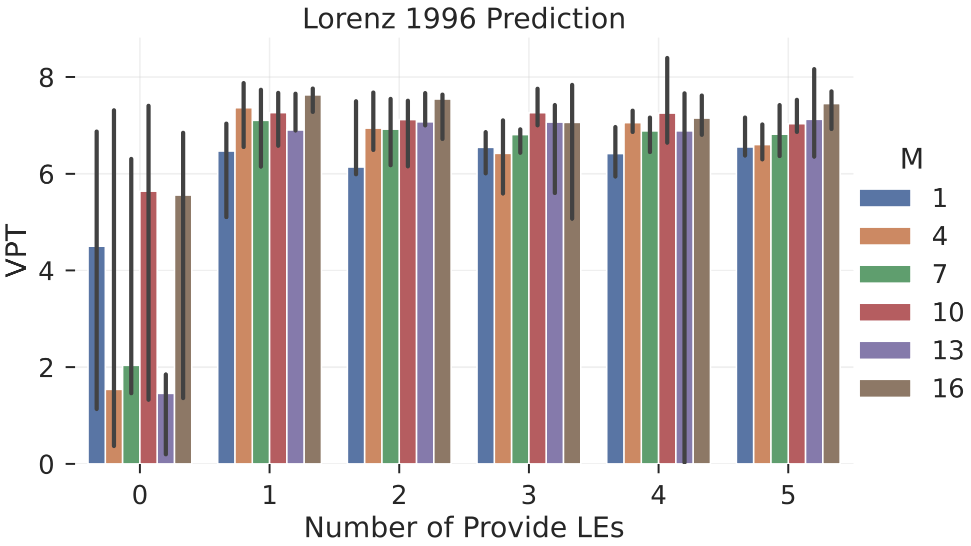

FIG. 8. This is the same data as in Fig.(5) for the 10 dimensional Lorenz 1996 system eq.(7) with F=8. Both the number of trajectories M, eq.(6), and the number of exponents provided to the cost function are varied, showing that the optimizer often fails to find a predictive solution when the number of provided exponents is 0. Even as the number of trajectories increases the mean predictive time over the sampled trajectories is extremely volatile. The procedure is that the optimizer is initialized 5 times for each condition and then the mean predictive time over the test set is recorded. Distributions are over these 5 points with the error bars giving the 95% confidence intervals.

with many provided trajectories, with no added dynamical information it is difficult to locate the correct minimum. Figures (9, 10) show how the found parameters change with the added information. There is a clear optimal spectral radius around ~ 0.2 that the RC without LEs fails to find. Additionally we show that the no LE case also fails to reproduce the correct largest LE, a clear sign of suboptimality[46].

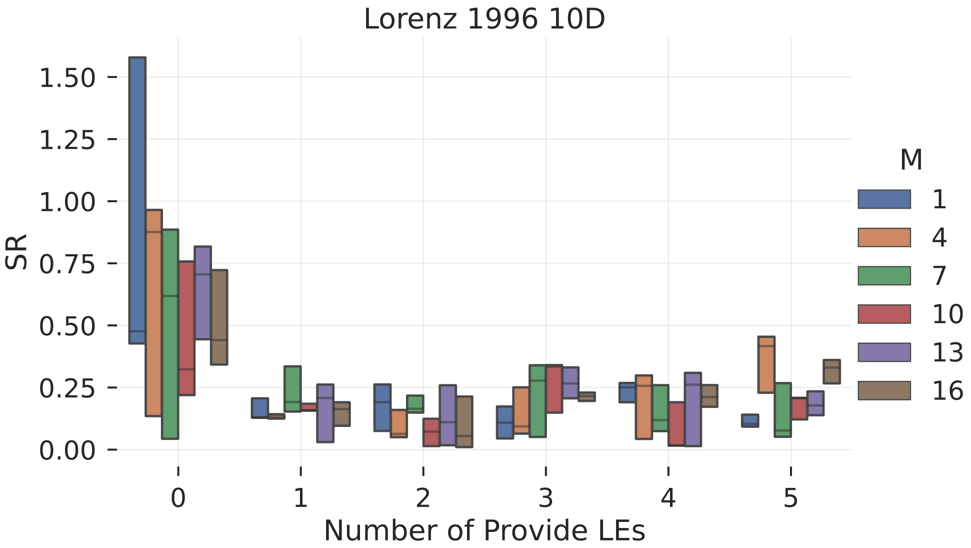

FIG. 9. Distribution of the spectral radii for the optimized RCs of the Lorenz 1996 system—the same as Fig.(8). The optimal SR is found by the optimizer when we provide the correct number of exponents but with 0 LEs provided it is unable to find the correct value.

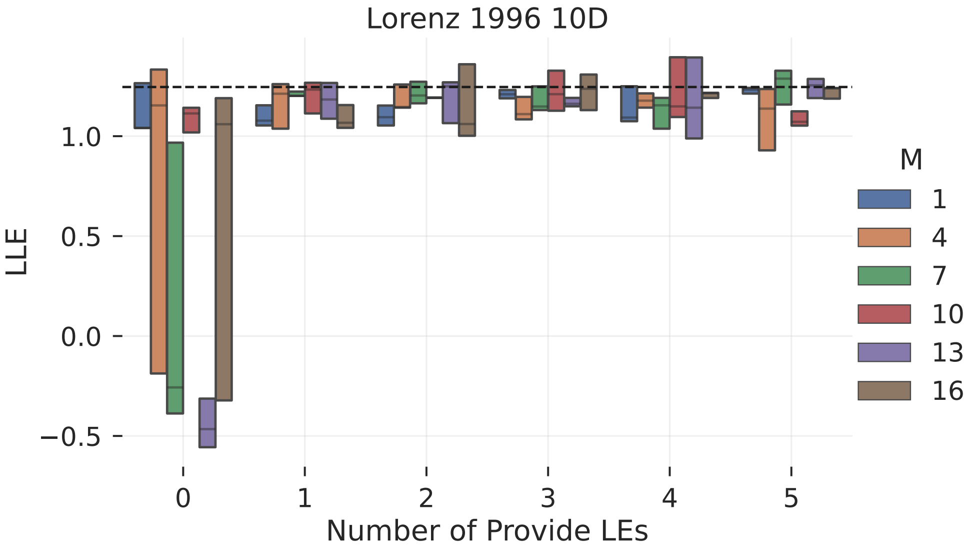

FIG. 10. Distribution of the largest LE for the optimized RCs of the Lorenz 1996 system—the same as Fig.(8). The correct largest LE (black line) is found by the optimizer when we provide LEs as inputs but with 0 LEs provided it is unable to find the correct value.