

Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model

Cite as: Chaos **28**, 041101 (2018); <https://doi.org/10.1063/1.5028373>

Submitted: 09 March 2018 . Accepted: 03 April 2018 . Published Online: 18 April 2018

Jaideep Pathak , Alexander Wikner, Rebeckah Fussell, Sarthak Chandra , Brian R. Hunt, Michelle Girvan, and Edward Ott

COLLECTIONS

 This paper was selected as Featured



View Online



Export Citation



CrossMark

ARTICLES YOU MAY BE INTERESTED IN

[Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data](#)

Chaos: An Interdisciplinary Journal of Nonlinear Science **27**, 121102 (2017); <https://doi.org/10.1063/1.5010300>

[Reservoir observers: Model-free inference of unmeasured variables in chaotic systems](#)

Chaos: An Interdisciplinary Journal of Nonlinear Science **27**, 041102 (2017); <https://doi.org/10.1063/1.4979665>

[Observing spatio-temporal dynamics of excitable media using reservoir computing](#)

Chaos: An Interdisciplinary Journal of Nonlinear Science **28**, 043118 (2018); <https://doi.org/10.1063/1.5022276>



AIP | Author Services

Learn more today!

Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model

Jaideep Pathak,¹ Alexander Wikner,² Rebeckah Fussell,³ Sarthak Chandra,¹ Brian R. Hunt,¹ Michelle Girvan,¹ and Edward Ott¹

¹University of Maryland, College Park, Maryland 20742, USA

²Rice University, Houston, Texas 77005, USA

³Haverford College, Haverford, Pennsylvania 19041, USA

(Received 9 March 2018; accepted 3 April 2018; published online 18 April 2018)

A model-based approach to forecasting chaotic dynamical systems utilizes knowledge of the mechanistic processes governing the dynamics to build an approximate mathematical model of the system. In contrast, machine learning techniques have demonstrated promising results for forecasting chaotic systems purely from past time series measurements of system state variables (training data), without prior knowledge of the system dynamics. The motivation for this paper is the potential of machine learning for filling in the gaps in our underlying mechanistic knowledge that cause widely-used knowledge-based models to be inaccurate. Thus, we here propose a general method that leverages the advantages of these two approaches by combining a knowledge-based model and a machine learning technique to build a hybrid forecasting scheme. Potential applications for such an approach are numerous (e.g., improving weather forecasting). We demonstrate and test the utility of this approach using a particular illustrative version of a machine learning known as reservoir computing, and we apply the resulting hybrid forecaster to a low-dimensional chaotic system, as well as to a high-dimensional spatiotemporal chaotic system. These tests yield extremely promising results in that our hybrid technique is able to accurately predict for a much longer period of time than either its machine-learning component or its model-based component alone. *Published by AIP Publishing.* <https://doi.org/10.1063/1.5028373>

Prediction of dynamical system states (e.g., as in weather forecasting) is a common and essential task with many applications in science and technology. This task is often carried out via a system of dynamical equations derived to model the process to be predicted. Due to deficiencies in knowledge or computational capacity, application of these models will generally be imperfect and may give unacceptably inaccurate results. On the other hand, data-driven methods, independent of the derived knowledge of the system, may be computationally intensive and may require unreasonably large amounts of data. In this paper, we consider a particular hybridization technique for combining these two approaches. Our tests of this hybrid technique suggest that it can be extremely effective and widely applicable.

extensive past measurements of the system state evolution (training data). Because the latter approach typically makes little or no use of mechanistic understanding, the amount of training data and the computational resources required can be prohibitive, especially when the system to be predicted is large and complex. The purpose of this paper is to propose, describe, and test a general framework for combining a knowledge-based approach with a machine learning approach to build a hybrid prediction scheme with significantly enhanced potential for performance and feasibility of implementation as compared to either an approximate knowledge-based model acting alone or a machine learning model acting alone. The results of our tests of our proposed hybrid scheme suggest that it can have wide applicability for forecasting in many areas of science and technology.

Another view motivating our hybrid approach is that, when trying to predict the evolution of a system, one might intuitively expect the best results when making appropriate use of *all* the available information about the system. Here, we think of both the (perhaps imperfect) knowledge-based model and the past measurement data as being two types of system information which we wish to simultaneously and efficiently utilize. Our hybrid technique combines the information in a way that does not presume either type to have greater predictive power. We note that hybrids of machine learning with other approaches have previously been applied to a variety of other tasks, but here, we consider the general problem of forecasting a dynamical system with an imperfect knowledge-based model, the form of whose imperfections is

I. INTRODUCTION

One approach to forecasting the state of a dynamical system starts by using whatever knowledge and understanding is available about the mechanisms governing the dynamics to construct a mathematical model of the system. Following that, one can use measurements of the system state to estimate initial conditions for the model that can then be integrated forward in time to produce forecasts. We refer to this approach as knowledge-based prediction. The accuracy of knowledge-based prediction is limited by any errors in the mathematical model. Another approach that has recently proven effective is to use machine learning to construct a predictor purely from

unknown. Examples of such other tasks addressed by machine learning hybrids include network anomaly detection,¹ credit rating,² and chemical process modeling,³ among others. We also note a recent preprint⁴ that employs a hybrid dynamical state forecasting scheme that is related to the one we present here.

To illustrate the hybrid scheme, we focus on a particular type of machine learning known as “reservoir computing,”^{5–7} which has been previously applied to the prediction of low dimensional systems⁸ and, more recently, to the prediction of large spatiotemporal chaotic systems.^{9,10} We emphasize that, while our illustration is for reservoir computing with a reservoir based on an artificial neural network, we view the results as a general test of the hybrid approach. As such, these results should be relevant to other versions of machine learning¹¹ (such as Long Short-Term Memory networks¹²), as well as reservoir computers in which the reservoir is implemented by various physical means (e.g., electro-optical schemes^{13–15} or Field Programmable Gate Arrays¹⁶). A particularly dramatic example illustrating the effectiveness of the hybrid approach is shown in Figs. 7(d)–7(f) in which, when acting alone, both the knowledge-based predictor and the reservoir machine learning predictor give fairly worthless results (prediction time of only a fraction of a Lyapunov time), but, when the same two systems are combined in the hybrid scheme, good predictions are obtained for a substantial duration of about 4 Lyapunov times. (By a “Lyapunov time” we mean the typical time required for an e -fold increase in the distance between two initially close chaotic orbits; see Secs. IV and V).

The rest of this paper is organized as follows: in Sec. II, we provide an overview of our methods for prediction by using a knowledge-based model and for prediction by exclusively using a reservoir computing approach (henceforth referred to as the reservoir-only predictor). We then describe the hybrid scheme that combines the knowledge-based model with the reservoir-only predictor. In Sec. III, we describe our specific implementation of the reservoir computing scheme and the proposed hybrid scheme using a recurrent-neural-network implementation of the reservoir computer. In Secs. IV and V, we demonstrate our hybrid prediction approach using two examples, namely, the low-dimensional Lorenz system¹⁷ and the high dimensional, spatiotemporal chaotic Kuramoto-Sivashinsky (KS) system.^{18,19}

II. PREDICTION METHODS

We consider a dynamical system for which there is available a time series of a set of M measurable state-dependent quantities, which we represent as an M dimensional vector $\mathbf{u}(t)$. As discussed earlier, we propose a hybrid scheme to make predictions of the dynamics of the system by combining an approximate knowledge-based prediction via an approximate model with a purely data-driven prediction scheme that uses machine learning. We will compare predictions made using our hybrid scheme with the predictions of the approximate knowledge-based model alone and predictions made by exclusively using the reservoir computing model.

A. Knowledge-based model

We obtain predictions from the approximate knowledge-based model acting alone assuming that the knowledge-based model is capable of forecasting $\mathbf{u}(t)$ for $t > 0$ based on an initial condition $\mathbf{u}(0)$ and possibly recent values of $\mathbf{u}(t)$ for $t < 0$. For notational use in our hybrid scheme (Sec. II C), we denote integration of the knowledge-based model forward in time by a time duration Δt as

$$\mathbf{u}_K(t + \Delta t) = \mathcal{K}[\mathbf{u}(t)] \approx \mathbf{u}(t + \Delta t). \quad (1)$$

We emphasize that the knowledge-based one-step-ahead predictor \mathcal{K} is imperfect and may have substantial unwanted error. In our test examples in Secs. IV and V, we consider prediction of continuous-time systems and take the prediction system time step Δt to be small compared to the typical time scale over which the continuous-time system changes. We note that while a single prediction time step (Δt) is small, we are interested in predicting for a large number of time steps.

B. Reservoir-only predictor

For the machine learning approach, we assume the knowledge of $\mathbf{u}(t)$ for times t from $-T$ to 0 . This data will be used to train the machine learning model for the purpose of making predictions of $\mathbf{u}(t)$ for $t > 0$. In particular, we use a reservoir computer, described as follows.

A reservoir computer (Fig. 1) is constructed with an artificial high dimensional dynamical system, known as the reservoir whose state is represented by the D_r dimensional

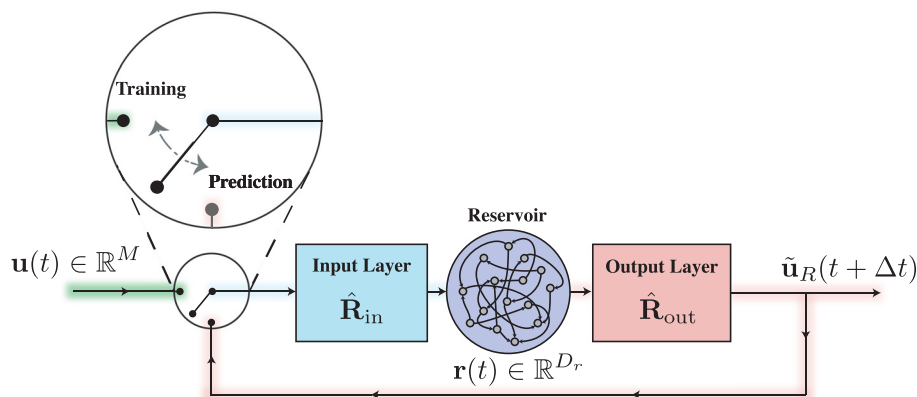


FIG. 1. Schematic diagram of the reservoir-only prediction setup.

vector $\mathbf{r}(t)$, $D_r \gg M$. We note that ideally the forecasting accuracy of a reservoir-only prediction increases with D_r , but that D_r is typically limited by computational cost considerations. The reservoir is coupled to an input through an Input-to-Reservoir coupling $\hat{\mathbf{R}}_{\text{in}}[\mathbf{u}(t)]$ which maps the M -dimensional input vector, \mathbf{u} , at time t , to each of the D_r reservoir state variables. The output is defined through a Reservoir-to-Output coupling $\hat{\mathbf{R}}_{\text{out}}[\mathbf{r}(t), \mathbf{p}]$, where \mathbf{p} is a large set of *adjustable* parameters. In the task of prediction of state variables of dynamical systems, the reservoir computer is used in two different configurations. One of the configurations we call the “training” phase, and the other one we called the “prediction” phase. In the training phase, the reservoir system is configured according to Fig. 1 with the switch in the position labeled “Training.” In this phase, the reservoir state evolves from $t = -T$ to $t = 0$ according to the equation

$$\mathbf{r}(t + \Delta t) = \hat{\mathbf{G}}_R[\hat{\mathbf{R}}_{\text{in}}[\mathbf{u}(t)], \mathbf{r}(t)], \quad (2)$$

where the nonlinear function $\hat{\mathbf{G}}_R$ and the (usually linear) function $\hat{\mathbf{R}}_{\text{in}}$ depend on the choice of the reservoir implementation. Next, we make a particular choice of the parameters \mathbf{p} such that the output function $\hat{\mathbf{R}}_{\text{out}}[\mathbf{r}(t), \mathbf{p}]$ satisfies

$$\hat{\mathbf{R}}_{\text{out}}[\mathbf{r}(t), \mathbf{p}] \approx \mathbf{u}(t),$$

for $-T < t \leq 0$. We achieve this by minimizing the error between $\tilde{\mathbf{u}}_R(t) = \hat{\mathbf{R}}_{\text{out}}[\mathbf{r}(t), \mathbf{p}]$ and $\mathbf{u}(t)$ for $-T < t \leq 0$ using a suitable error metric and optimization algorithm on the adjustable parameter vector \mathbf{p} .

In the prediction phase, for $t \geq 0$, the switch is placed in position labeled “Prediction” indicated in Fig. 1. The reservoir state now evolves autonomously with a feedback loop according to the equation

$$\mathbf{r}(t + \Delta t) = \hat{\mathbf{G}}_R[\hat{\mathbf{R}}_{\text{in}}[\tilde{\mathbf{u}}_R(t)], \mathbf{r}(t)], \quad (3)$$

where $\tilde{\mathbf{u}}_R(t) = \hat{\mathbf{R}}_{\text{out}}[\mathbf{r}(t), \mathbf{p}]$ is taken as the prediction from this reservoir-only approach. It has been shown⁸ that this procedure can successfully generate a time series $\tilde{\mathbf{u}}_R(t)$ that approximates the true state $\mathbf{u}(t)$ for $t > 0$. Thus, $\tilde{\mathbf{u}}_R(t)$ is our reservoir-based prediction of the evolution of $\mathbf{u}(t)$. If, as assumed henceforth, the dynamical system being predicted is chaotic, the exponential divergence of initial conditions in the dynamical system implies that any prediction scheme

will only be able to yield an accurate prediction for a limited amount of time.

C. Hybrid scheme

The hybrid approach we propose combines both the knowledge-based model and the reservoir-only predictor. Our hybrid approach is outlined in the schematic diagram shown in Fig. 2.

As in the reservoir-only approach, the hybrid scheme has two phases, the training phase and the prediction phase. In the training phase (with the switch in position labeled “Training” in Fig. 2), the training data $\mathbf{u}(t)$ from $t = -T$ to $t = 0$ is fed into both the knowledge-based predictor and the reservoir. At each time t , the output of the knowledge-based predictor is the one-step ahead prediction $\mathcal{K}[\mathbf{u}(t)]$. The reservoir evolves according to the equation

$$\mathbf{r}(t + \Delta t) = \hat{\mathbf{G}}_H[\mathbf{r}(t), \hat{\mathbf{H}}_{\text{in}}[\mathcal{K}[\mathbf{u}(t)], \mathbf{u}(t)]] \quad (4)$$

for $-T \leq t \leq 0$, where the (usually linear) function $\hat{\mathbf{H}}_{\text{in}}$ couples the reservoir network with the inputs to the reservoir, in this case, $\mathbf{u}(t)$ and $\mathcal{K}[\mathbf{u}(t)]$. As earlier, we modify a set of adjustable parameters \mathbf{p} in a predefined output function so that

$$\hat{\mathbf{H}}_{\text{out}}[\mathcal{K}[\mathbf{u}(t - \Delta t)], \mathbf{r}(t), \mathbf{p}] \approx \mathbf{u}(t) \quad (5)$$

for $-T < t \leq 0$, which is achieved by minimizing the error between the right-hand side and the left-hand side of Eq. (5), as discussed earlier (Sec. II B) for the reservoir-only approach. Note that both the knowledge-based model and the reservoir feed into the output layer [Eq. (5) and Fig. 2] so that the training can be thought of as optimally deciding on how to weight the information from the knowledge-based and reservoir components.

For the prediction phase (the switch is placed in the position labeled “Prediction” in Fig. 2), the feedback loop is closed allowing the system to evolve autonomously. The dynamics of the system will then be given by

$$\mathbf{r}(t + \Delta t) = \hat{\mathbf{G}}_H[\mathbf{r}(t), \hat{\mathbf{H}}_{\text{in}}[\mathcal{K}[\tilde{\mathbf{u}}_H(t)], \tilde{\mathbf{u}}_H(t)]], \quad (6)$$

where $\tilde{\mathbf{u}}_H(t) = \hat{\mathbf{H}}_{\text{out}}[\mathcal{K}[\tilde{\mathbf{u}}_H(t - \Delta t)], \mathbf{r}(t), \mathbf{p}]$ is the prediction of the prediction of the hybrid system.

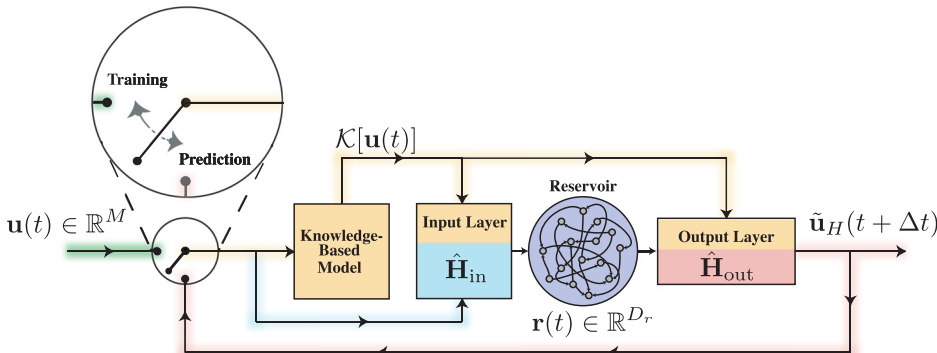


FIG. 2. Schematic diagram of the hybrid prediction setup.

III. IMPLEMENTATION

In this section, we provide details of our specific implementation and discuss the prediction performance metrics we use to assess and compare the various prediction schemes. Our implementation of the reservoir computer closely follows Ref. 8. Note that, in the reservoir training, no knowledge of the dynamics and details of the reservoir (the network within the circles in Figs. 1 and 2) is used (this contrasts with other machine learning techniques¹¹): only the $-T \leq t \leq 0$ training data is used ($\mathbf{u}(t)$, $\mathbf{r}(t)$, and, in the case of the hybrid, $\mathcal{K}[\mathbf{u}(t)]$). This feature implies that reservoir computers, as well as the reservoir-based hybrid are insensitive to the specific reservoir implementation. In this paper, our illustrative implementation of the reservoir computer uses an artificial neural network for the realization of the reservoir. We mention, however, that alternative implementation strategies such as utilizing nonlinear optical devices^{13–15} and Field Programmable Gate Arrays¹⁶ can also be used to construct the reservoir component of our hybrid scheme (Fig. 2) and offer potential advantages, particularly with respect to speed.

A. Reservoir-only and hybrid implementations

Here, we consider that the high-dimensional reservoir is implemented by a large, weighted, directed, low degree Erdős-Rényi network of D_r nonlinear, neuron-like units in which the network is described by an adjacency matrix \mathbf{A} (we stress that the following implementations are somewhat arbitrary, and are intended as illustrating typical results that might be expected). The network is constructed to have an average degree denoted by $\langle d \rangle$, and the nonzero elements of \mathbf{A} , representing the edge weights in the network, are initially chosen independently from the uniform distribution over the interval $[-1, 1]$. All the edge weights in the network are then uniformly scaled via multiplication of the adjacency matrix by a constant factor to set the largest magnitude eigenvalue of the matrix to a quantity ρ , which is called the “spectral radius” of \mathbf{A} . The state of the reservoir, given by the vector $\mathbf{r}(t)$, consists of the components r_j for $1 \leq j \leq D_r$ where $r_j(t)$ denotes the scalar state of the j th node in the network. When evaluating prediction based purely on a reservoir system alone, the reservoir is coupled to the M dimensional input through a $D_r \times M$ dimensional matrix \mathbf{W}_{in} , such that in Eq. (2) $\hat{\mathbf{R}}_{\text{in}}[\mathbf{u}(t)] = \mathbf{W}_{\text{in}}\mathbf{u}(t)$, and each row of the matrix \mathbf{W}_{in} has exactly one randomly chosen nonzero element. Each nonzero element of the matrix is independently chosen from the uniform distribution on the interval $[-\sigma, \sigma]$. We choose the hyperbolic tangent function for the form of the nonlinearity at the nodes, so that the specific training phase equation for our reservoir setup corresponding to Eq. (2) is

$$\mathbf{r}(t + \Delta t) = \tanh[\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{\text{in}}\mathbf{u}(t)], \quad (7)$$

where the hyperbolic tangent applied on a vector is defined as the vector whose components are the hyperbolic tangent function acting on each element of the argument vector individually.

We choose the form of the output function to be $\hat{\mathbf{R}}_{\text{out}}(\mathbf{r}, \mathbf{p}) = \mathbf{W}_{\text{out}}\mathbf{r}^*$, in which the output parameters (previously symbolically represented by \mathbf{p}) will henceforth be taken to be the elements of the matrix \mathbf{W}_{out} , and the vector \mathbf{r}^* is defined such that r_j^* equals r_j for odd j , and equals r_j^2 for even j (it was empirically found that this choice of \mathbf{r}^* works well for our examples in both Secs. IV and V, see also Refs. 10 and 20). We run the reservoir for $-T \leq t \leq 0$ with the switch in Fig. 1 in the “Training” position. We then use Tikhonov regularized linear regression²¹ to train $\tilde{\mathbf{u}}_R(t) = \mathbf{W}_{\text{out}}\mathbf{r}^*(t)$ to approximate $\mathbf{u}(t)$. That is, we minimize the quantity $\sum_{m=1}^{T/\Delta t} \|\mathbf{u}(-m\Delta t) - \tilde{\mathbf{u}}_R(-m\Delta t)\|^2 + \beta \|\mathbf{W}_{\text{out}}\|^2$, where $\|\mathbf{W}_{\text{out}}\|^2$ is the sum of the squares of the matrix elements of \mathbf{W}_{out} and the regularization parameter β is a small positive number introduced to avoid overfitting. Since $\tilde{\mathbf{u}}_R$ depends linearly on the elements of \mathbf{W}_{out} , this minimization is a standard linear regression problem.

Once the output parameters (the matrix elements of \mathbf{W}_{out}) are determined, we run the system in the configuration depicted in Fig. 1 with the switch in the “Prediction” position according to the equations

$$\tilde{\mathbf{u}}_R(t) = \mathbf{W}_{\text{out}}\mathbf{r}^*(t), \quad (8)$$

$$\mathbf{r}(t + \Delta t) = \tanh[\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{\text{in}}\tilde{\mathbf{u}}_R(t)], \quad (9)$$

corresponding to Eq. (3). Here, $\tilde{\mathbf{u}}_R(t)$ denotes the prediction of $\mathbf{u}(t)$ for $t > 0$ made by the reservoir-only model.

Next, we describe the implementation of the hybrid prediction scheme. The reservoir component of our hybrid scheme is implemented in the same fashion as in the reservoir-only model given above. In the training phase for $-T < t \leq 0$, when the switch in Fig. 2 is in the “Training” position, the specific form of Eq. (4) used is given by

$$\mathbf{r}(t + \Delta t) = \tanh \left[\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{\text{in}} \begin{pmatrix} \mathcal{K}[\mathbf{u}(t)] \\ \mathbf{u}(t) \end{pmatrix} \right]. \quad (10)$$

As earlier, we choose the matrix \mathbf{W}_{in} (which is now $D_r \times (2M)$ dimensional) to have exactly one nonzero element in each row. The nonzero elements are independently chosen from the uniform distribution on the interval $[-\sigma, \sigma]$. Each nonzero element can be interpreted to correspond to a connection to a particular reservoir node. These nonzero elements are randomly chosen such that a fraction γ of the reservoir nodes is connected exclusively to the raw input $\mathbf{u}(t)$ and the remaining fraction is connected exclusively to the output of the model based predictor $\mathcal{K}[\mathbf{u}(t)]$.

Similar to the reservoir-only case, we choose the form of the output function to be

$$\hat{\mathbf{H}}_{\text{out}}[\mathcal{K}[\mathbf{u}(t - \Delta t)], \mathbf{r}(t), \mathbf{p}] = \mathbf{W}_{\text{out}} \begin{pmatrix} \mathcal{K}[\mathbf{u}(t - \Delta t)] \\ \mathbf{r}^*(t) \end{pmatrix}, \quad (11)$$

where, as in the reservoir-only case, \mathbf{W}_{out} now plays the role of \mathbf{p} . Again, as in the reservoir-only case, \mathbf{W}_{out} is determined via Tikhonov regularized regression.

In the prediction phase for $t > 0$, when the switch in Fig. 2 is in position labeled “Prediction,” the input $\mathbf{u}(t)$ is replaced by the output at the previous time step and the equation analogous to Eq. (6) is given by

$$\tilde{\mathbf{u}}_H(t) = \mathbf{W}_{\text{out}} \begin{pmatrix} \mathcal{K}[\mathbf{u}(t)] \\ \mathbf{r}^*(t) \end{pmatrix}, \quad (12)$$

$$\mathbf{r}(t + \Delta t) = \tanh \left[\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{\text{in}} \begin{pmatrix} \mathcal{K}[\tilde{\mathbf{u}}_H] \\ \tilde{\mathbf{u}}_H \end{pmatrix} \right]. \quad (13)$$

The vector time series $\tilde{\mathbf{u}}_H(t)$ denotes the prediction of $\mathbf{u}(t)$ for $t > 0$ made by our hybrid scheme.

B. Training reusability

In the prediction phase, $t > 0$, chaos combined with a small initial condition error, $\|\tilde{\mathbf{u}}(0) - \mathbf{u}(0)\| \ll \|\mathbf{u}(0)\|$, and imperfect reproduction of the true system dynamics by the prediction method lead to a roughly exponential increase in the prediction error $\|\tilde{\mathbf{u}}(t) - \mathbf{u}(t)\|$ as the prediction time t increases. Eventually, the prediction error becomes unacceptably large. By choosing a value for the largest acceptable prediction error, one can define a “valid time” t_v for a particular trial. As our examples in Secs. IV and V show, t_v is typically much less than the necessary duration T of the training data required for either reservoir-only prediction or for prediction by our hybrid scheme. However, it is important to point out that the reservoir and hybrid schemes have the property of *training reusability*. That is, once the output parameters \mathbf{p} (or \mathbf{W}_{out}) are obtained using the training data in $-T \leq t \leq 0$, the same \mathbf{p} can be used over and over again for subsequent predictions, without retraining \mathbf{p} . For example, say that we now desire another prediction starting at some later time $t_0 > 0$. In order to do this, the reservoir system (Fig. 1) or the hybrid system (Fig. 2) with the predetermined \mathbf{p} , is first run with the switch in the “Training” position for a time, $(t_0 - \xi) < t < t_0$. This is done in order to resynchronize the reservoir state $\mathbf{r}(t_0)$ to the dynamics of the true system. Then, at time $t = t_0$, the switch (Figs. 1 and 2) is moved to the “Prediction” position, and the output $\tilde{\mathbf{u}}_R$ or $\tilde{\mathbf{u}}_H$ is taken as the prediction for $t > t_0$. We find that with \mathbf{p} predetermined, the time required for re-synchronization ξ turns out to be very small compared to t_v , which is in turn small compared to the training time T .

C. Assessments of prediction methods

We wish to compare the effectiveness of different prediction schemes (knowledge-based, reservoir-only, or hybrid). As previously mentioned, for each independent prediction, we quantify the duration of accurate prediction with the corresponding “valid time”, denoted t_v , defined as the elapsed time before the normalized error $E(t)$ first exceeds some value f , $0 < f < 1$, $E(t_v) = f$, where

$$E(t) = \frac{\|\mathbf{u}(t) - \tilde{\mathbf{u}}(t)\|}{\langle \|\mathbf{u}(t)\|^2 \rangle^{1/2}}, \quad (14)$$

and the symbol $\tilde{\mathbf{u}}(t)$ now stands for the prediction [either $\tilde{\mathbf{u}}_K(t)$, $\tilde{\mathbf{u}}_R(t)$, or $\tilde{\mathbf{u}}_H(t)$ as obtained by either of the three prediction methods (knowledge-based, reservoir-based, or hybrid)].

In what follows, we use $f = 0.4$. We test all three methods on 20 disjoint time intervals of length τ in a long run of

the true dynamical system. For each prediction method, we evaluate the valid time over many independent prediction trials. Further, for the reservoir-only prediction and the hybrid schemes, we use 32 different random realizations of \mathbf{A} and \mathbf{W}_{in} , for each of which we separately determine the training output parameters \mathbf{W}_{out} ; then we predict on each of the 20 time intervals for each such random realization, taking advantage of training reusability (Sec. III B). Thus, there are a total of 640 different trials for the reservoir-only and hybrid system methods, and 20 trials for the knowledge-based method. We use the median valid time across all such trials as a measure of the quality of prediction of the corresponding scheme, and the span between the first and third quartiles of the t_v values as a measure of variation in this metric of the prediction quality.

IV. LORENZ SYSTEM

The Lorenz system¹⁷ is described by the equations

$$\begin{aligned} \frac{dx}{dt} &= -ax + ay, \\ \frac{dy}{dt} &= bx - y - xz, \\ \frac{dz}{dt} &= -cz + xy. \end{aligned} \quad (15)$$

For our “true” dynamical system, we use $a = 10$, $b = 28$, and $c = 8/3$ which we use to generate simulated data in $-T \leq t \leq 0$ and to generate true orbits in $t > 0$ for comparison with our predictions. For our knowledge-based predictor, we use an “imperfect” version of the Lorenz equations to represent an approximate, imperfect model that might be encountered in a real life situation. Our imperfect model differs from the true Lorenz system given in Eq. (15) only via a change in the system parameter b in Eq. (15) to $b(1 + \epsilon)$. The error parameter ϵ is thus a dimensionless quantification of the discrepancy between our knowledge-based predictor and the “true” Lorenz system. We emphasize that, although we simulate model error by a shift of the parameter b , we view this to represent a general model error of *unknown form*. For example, we can view our parameter mismatch ϵ as a surrogate for a situation where the best available model differs by order ϵ from the dynamics due to factors such as imperfect knowledge basis of the system, too crude subgrid-scale modeling, etc. (e.g., see Ref. 4, where the knowledge-based system is a Galerkin approximation of a higher dimensional true system). This view is reflected by the fact that our reservoir and hybrid methods do not incorporate knowledge that the system error in our experiments results from an imperfect parameter value of a system with the Lorenz form. Next, for the reservoir computing component of the hybrid scheme, we construct a network-based reservoir as discussed in Sec. II B for various reservoir sizes D_r and with the parameters listed in Table I.

Figure 3 shows an illustrative example of one prediction trial using the hybrid method. The horizontal axis is the time in units of the Lyapunov time $\lambda_{\text{max}}^{-1}$, where λ_{max} denotes the largest Lyapunov exponent of the system [Eqs. (15)]. The vertical dashed lines in Fig. 3 indicate the valid time t_v

TABLE I. Reservoir parameters ρ , $\langle d \rangle$, σ , Δt , training time T , hybrid parameter γ , and evaluation parameters τ , ξ for the Lorenz system prediction.

Parameter	Value	Parameter	Value
P	0.4	T	100
$\langle d \rangle$	3	γ	0.5
σ	0.15	τ	250
Δt	0.1	ξ	10

(Sec. III C) at which $E(t)$ [Eq. (14)] first reaches the value $f=0.4$. The valid time determination for this example with $\epsilon=0.05$ and $D_r=500$ is illustrated in Fig. 4. Notice that we get a low prediction error for about 10 Lyapunov times.

The red upper curve in Fig. 5 shows the dependence on the reservoir size D_r of results for the median valid time (in units of Lyapunov time, $\lambda_{\max} t$, and with $f=0.4$) of the predictions from a hybrid scheme using a reservoir system combined with our imperfect model with an error parameter of $\epsilon=0.05$. The error bars span the first and third quartiles of our trials which are generated as described in Sec. III C. The black middle curve in Fig. 5 shows the corresponding results for predictions using the reservoir-only scheme. The blue lower curve in Fig. 5 shows the result for prediction using only the $\epsilon=0.05$ imperfect knowledge-based model (since this result does not depend on D_r , the blue curve is horizontal, and the error bars are the same at each value of D_r). Note that, even though the knowledge-based prediction alone is very bad, when used in the hybrid, it results in a large prediction improvement relative to the reservoir-only prediction. Moreover, this improvement is seen for all values of the reservoir sizes tested. Note also that the valid time for the hybrid with a reservoir size of $D_r=50$ is comparable to the valid time for a reservoir-only scheme at $D_r=500$. This suggests that our hybrid method can substantially reduce reservoir computational expense even with a knowledge-based model that has low predictive power on its own.

Figure 6 shows the dependence of prediction performance on the model error ϵ with the reservoir size held fixed

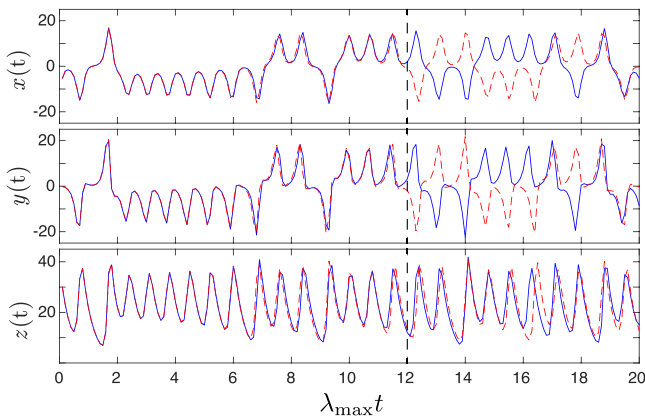


FIG. 3. Prediction of the Lorenz system using the hybrid prediction setup. The blue line shows the true state of the Lorenz system and the red dashed line shows the prediction. Prediction begins at $t=0$. The vertical black dashed line marks the point where this prediction is no longer considered valid by the valid time metric with $f=0.4$. The error in the approximate model used in the knowledge-based component of the hybrid scheme is $\epsilon=0.05$.

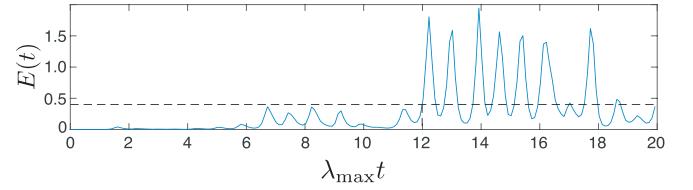


FIG. 4. Normalized error $E(t)$ versus time of the Lorenz prediction trial shown in Fig. 3. The prediction error remains below the defined threshold ($E(t) < 0.4$) for about 12 Lyapunov times.

at $D_r=50$. For the wide range of the error ϵ we have tested, the hybrid performance is much better than either its knowledge-based component alone or the reservoir-only component. Figures 5 and 6, taken together, suggest the potential robustness of the utility of the hybrid approach.

V. KURAMOTO-SIVASHINSKY EQUATIONS

In this example, we test how well our hybrid method, using an inaccurate knowledge-based model combined with a relatively small reservoir, can predict systems that exhibit high dimensional spatiotemporal chaos. Specifically, we use simulated data from the one-dimensional Kuramoto-Sivashinsky (KS) equation for $y(x, t)$

$$y_t = -yy_x - y_{xx} - y_{xxx}. \quad (16)$$

Our simulation calculates $y(x, t)$ on a uniformly spaced grid with spatially periodic boundary conditions such that $y(x, t) = y(x + L, t)$, with a periodicity length of $L = 35$, a grid size of $Q = 64$ grid points (giving a intergrid spacing of $\Delta x = \frac{L}{Q} \approx 0.547$), and a sampling time of $\Delta t = 0.25$. For these parameters, we found that the maximum Lyapunov exponent, λ_{\max} , is positive ($\lambda_{\max} \approx 0.07$), indicating that this system is chaotic. We define a vector of $y(x, t)$ values at each grid point as the input to each of our predictors

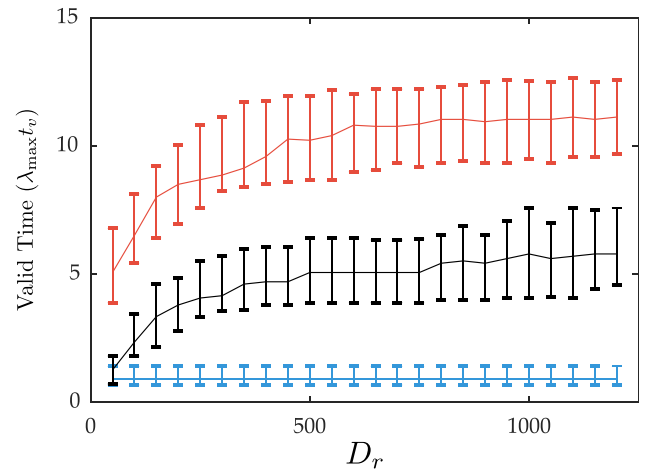


FIG. 5. Reservoir size (D_r) dependence of the median valid time using the hybrid prediction scheme (red upper plot), the reservoir-only (black middle plot), and the knowledge-based model only methods. The model error is fixed at $\epsilon=0.05$. Since the knowledge based model (blue) does not depend on D_r , its plot is a horizontal line. The error bars span the range between the 1st and 3rd quartiles of the trials.

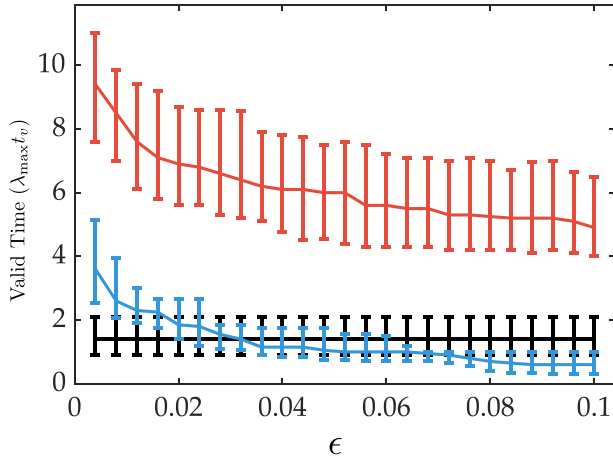


FIG. 6. Valid times for different values of model error (ϵ) with $f=0.4$. The reservoir size is fixed at $D_r=50$. The plotted points represent the median and the error bars span the range between the 1st and 3rd quartiles. The meaning of the colors is the same as in Fig. 5. Since the reservoir only scheme (black) does not depend on ϵ , its plot is a horizontal line. Similar to Fig. 5, the small reservoir alone cannot predict well for a long time, but the hybrid model, which combines the inaccurate knowledge-based model and the small reservoir performs well across a broad range of ϵ .

$$\mathbf{u}(t) = \left[y\left(\frac{L}{Q}, t\right), y\left(\frac{2L}{Q}, t\right), \dots, y(L, t) \right]^T. \quad (17)$$

For our approximate knowledge-based predictor, we use the same simulation method as the original Kuramoto-Sivashinsky equations with an error parameter ϵ added to the coefficient of the second derivative term as follows:

$$y_t = -yy_x - (1 + \epsilon)y_{xx} - y_{xxx}. \quad (18)$$

For sufficiently small ϵ , Eq. (18) corresponds to a very accurate knowledge-based model of the true KS system, which becomes less and less accurate as ϵ is increased.

Illustrations of our main result are shown in Figs. 7 and 8, where we use the parameters in Table II. In the top panel of Fig. 7, we plot a computed solution of Eq. (16) which we regard as the true dynamics of a system to be predicted; the spatial coordinate $x \in [0, L]$ is plotted vertically, the time in Lyapunov units ($\lambda_{\max} t$) is plotted horizontally, and the value of $y(x, t)$ is color coded with the most positive and most negative y values indicated by red and blue, respectively. Below this top panel are six panels labeled (a-f) in which the color coded quantity is the prediction error $\tilde{y}(x, t) - y(x, t)$ of different predictions $\tilde{y}(x, t)$. In panels (a), (b), and (c), we consider a case ($\epsilon=0.01$, $D_r=8000$) where both the knowledge-based model [panel (a)] and the reservoir-only predictor [panel (b)] are fairly accurate; panel (c) shows the hybrid prediction error. In panels (d), (e), and (f), we consider a different case ($\epsilon=0.1$, $D_r=500$) where both the knowledge-based model [panel (d)] and the reservoir-only predictor [panel (e)] are relatively inaccurate; panel (f) shows the hybrid prediction error. In our color coding, the low prediction error corresponds to the green color. The vertical solid lines denote the valid times for this run with $f=0.4$.

We note from Figs. 7(a) to 7(c), that when the knowledge-based model prediction is valid for about as long

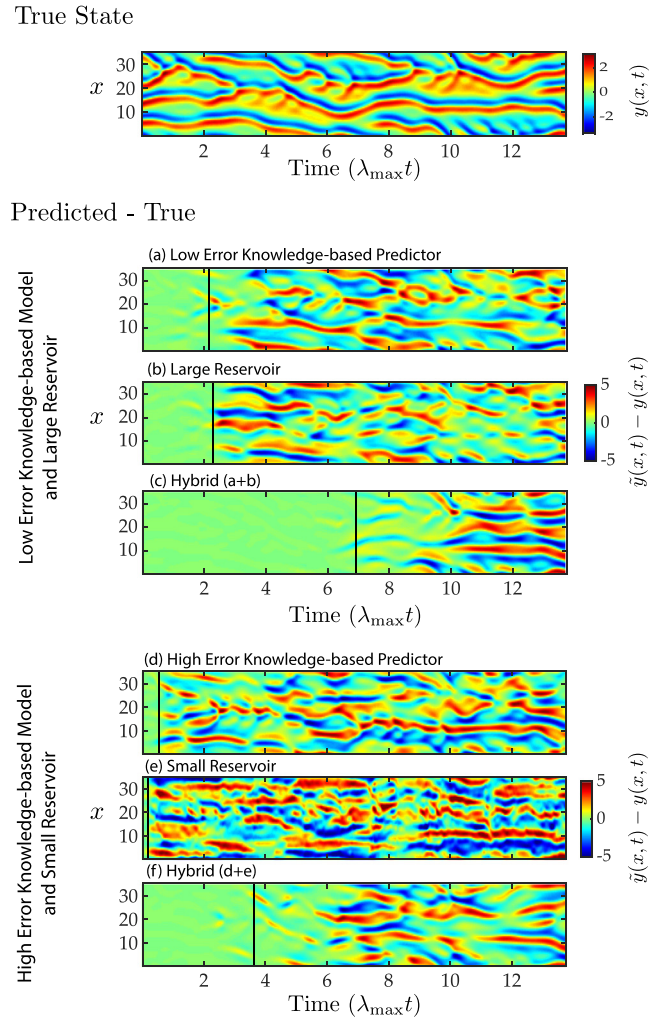


FIG. 7. The topmost panel shows the true solution of the KS equation [Eq. (16)]. Each of the six panels labeled (a) through (f) shows the difference between the true state of the KS system and the prediction made by a specific prediction scheme. The three panels (a), (b), and (c), respectively, show the results for a low error knowledge-based model ($\epsilon=0.01$), a reservoir-only prediction scheme with a large reservoir ($D_r=8000$), and the hybrid scheme composed of ($D_r=8000$, $\epsilon=0.01$). The three panels, (d), (e), and (f), respectively, show the corresponding results for a highly imperfect knowledge-based model ($\epsilon=0.1$), a reservoir-only prediction scheme using a small reservoir ($D_r=500$), and the hybrid scheme with ($D_r=500$, $\epsilon=0.1$).

as the reservoir-only prediction, our hybrid scheme can significantly outperform both its components. Additionally, as in our Lorenz system example (Fig. 6 for $\epsilon \geq 0.2$), we see from Figs. 7(d) to 7(f) that in the parameter regimes where the KS reservoir-only predictor and knowledge-based model both show very poor performance, the hybrid of these low performing methods can still predict for a longer time than a much more computationally expensive reservoir-only implementation [Fig. 7(b)].

This latter remarkable result is reinforced by Fig. 8(a), which shows that even for a very large error, $\epsilon=1$, such that the model is totally ineffective, the hybrid of these two methods is able to predict for a significant amount of time using a relatively small reservoir. This implies that a non-viable model can be made viable via the addition of a reservoir component of modest size. Further, Figs. 8(b) and 8(c) show that even if one has a model that can outperform the

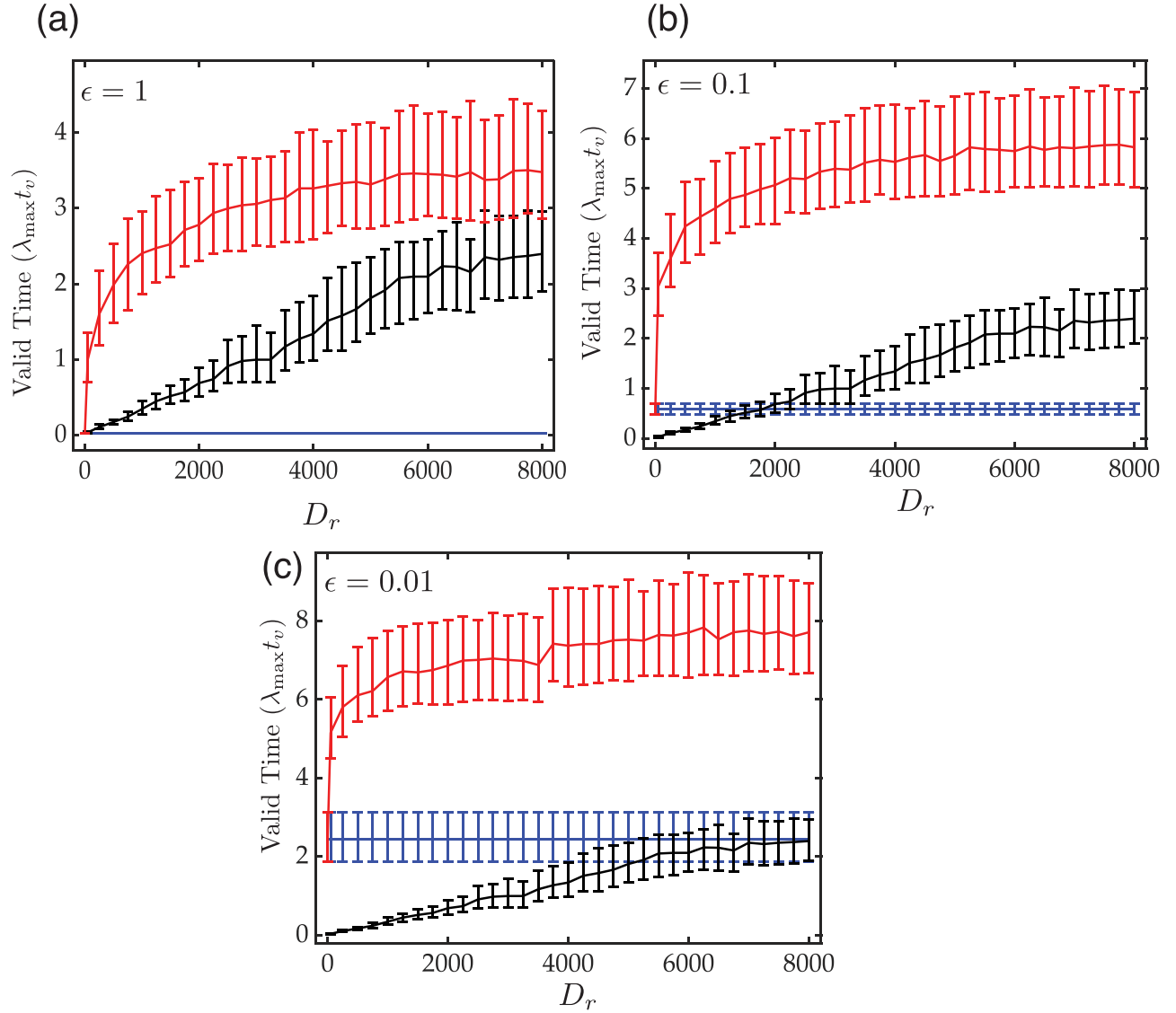


FIG. 8. Each of the three panels (a), (b), and (c) shows a comparison of the KS system prediction performance of the reservoir-only scheme (black), the knowledge-based model (blue), and the hybrid scheme (red). The median valid time in Lyapunov units ($\lambda_{\max} t_v$) is plotted against the size of the reservoir used in the hybrid scheme and the reservoir-only scheme. Since the knowledge-based model does not use a reservoir, its valid time does not vary with the reservoir size. The error in the knowledge-based model is $\epsilon = 1$ in panel (a), $\epsilon = 0.1$ in panel (b), and $\epsilon = 0.01$ in panel (c).

reservoir prediction, as is the case for $\epsilon = 0.01$ for most reservoir sizes, one can still benefit from a reservoir using our hybrid technique.

VI. CONCLUSIONS

In this paper, we present a method for the prediction of chaotic dynamical systems that hybridizes reservoir computing and knowledge-based prediction. Our main results are as follows:

TABLE II. Reservoir parameters ρ , $\langle d \rangle$, σ , Δt , training time T , hybrid parameter γ , and evaluation parameters τ , ξ for the KS system prediction.

Parameter	Value	Parameter	Value
P	0.4	T	5000
$\langle d \rangle$	3	γ	0.5
σ	1.0	τ	250
Δt	0.25	ξ	10

- (1) Our hybrid technique consistently outperforms its component reservoir-only or knowledge-based model prediction methods in the duration of its ability to accurately predict, for both the Lorenz system and the spatiotemporal chaotic Kuramoto-Sivashinsky equations.
- (2) Our hybrid technique robustly yields improved performance even when the reservoir-only predictor and the knowledge-based model are so flawed that they do not make accurate predictions on their own.
- (3) Even when the knowledge-based model used in the hybrid is significantly flawed, the hybrid technique can, at small reservoir sizes, make predictions comparable to those made by much larger reservoir-only predictors, which can be used to save computational resources.
- (4) Both the hybrid scheme and the reservoir-only predictor have the property of “training reusability” (Sec. III B), meaning that once trained, they can make any number of subsequent predictions (without retraining each time) by

preceding each such prediction with a short run in the training configuration (see Figs. 1 and 2) in order to resynchronize the reservoir dynamics with the dynamics to be predicted.

ACKNOWLEDGMENTS

This work was supported by ARO (W911NF-12-1-0101), NSF (PHY-1461089), and DARPA.

- ¹T. Shon and J. Moon, “A hybrid machine learning approach to network anomaly detection,” *Inf. Sci.* **177**, 3799–3821 (2007).
- ²C.-F. Tsai and M.-L. Chen, “Credit rating by hybrid machine learning techniques,” *Appl. Soft Comput.* **10**, 374–380 (2010).
- ³D. C. Psichogios and L. H. Ungar, “A hybrid neural network-first principles approach to process modeling,” *AIChE J.* **38**, 1499–1511 (1992).
- ⁴Z. Y. Wan, P. R. Vlachas, P. Koumoutsakos, and T. P. Sapsis, “Data-assisted reduced-order modeling of extreme events in complex dynamical systems,” preprint [arXiv:1803.03365v1](https://arxiv.org/abs/1803.03365v1) (2018); This work employs Long Short-Term Memory networks¹² rather than reservoir computing.
- ⁵H. Jaeger, “The “echo state” approach to analyzing and training recurrent neural networks—with an erratum note,” GMD Technical Report No. 148, German National Research Center for Information Technology, Bonn, Germany, 2001.
- ⁶W. Maass, T. Natschläger, and H. Markram, “Real-time computing without stable states: A new framework for neural computation based on perturbations,” *Neural Comput.* **14**, 2531–2560 (2002).
- ⁷M. Lukoševičius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Comput. Sci. Rev.* **3**, 127–149 (2009).
- ⁸H. Jaeger and H. Haas, “Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication,” *Science* **304**, 78–80 (2004).
- ⁹J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, “Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data,” *Chaos* **27**, 121102 (2017).
- ¹⁰J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, “Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach,” *Phys. Rev. Lett.* **120**, 024102 (2018).
- ¹¹I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).
- ¹²S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.* **9**, 1735–1780 (1997).
- ¹³L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutiérrez, L. Pesquera, C. R. Mirasso, and I. Fischer, “Photonic information processing beyond Turing: An optoelectronic implementation of reservoir computing,” *Opt. Express* **20**, 3241–3249 (2012).
- ¹⁴L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, “High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification,” *Phys. Rev. X* **7**, 011015 (2017).
- ¹⁵P. Antonik, M. Haelterman, and S. Massar, “Brain-inspired photonic signal processor for generating periodic patterns and emulating chaotic systems,” *Phys. Rev. Appl.* **7**, 054014 (2017).
- ¹⁶N. D. Haynes, M. C. Soriano, D. P. Rosin, I. Fischer, and D. J. Gauthier, “Reservoir computing with a single time-delay autonomous Boolean node,” *Phys. Rev. E* **91**, 020801 (2015).
- ¹⁷E. N. Lorenz, “Deterministic nonperiodic flow,” *J. Atmos. Sci.* **20**, 130–141 (1963).
- ¹⁸Y. Kuramoto and T. Tsuzuki, “Persistent propagation of concentration waves in dissipative media far from thermal equilibrium,” *Prog. Theor. Phys.* **55**, 356–369 (1976).
- ¹⁹G. Sivashinsky, “Nonlinear analysis of hydrodynamic instability in laminar flames—i. derivation of basic equations,” *Acta Astronaut.* **4**, 1177–1206 (1977).
- ²⁰Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, and E. Ott, “Reservoir observers: Model-free inference of unmeasured variables in chaotic systems,” *Chaos* **27**, 041102 (2017).
- ²¹A. N. Tikhonov, V. I. Arsenin, and F. John, *Solutions of Ill-Posed Problems* (Winston, Washington, DC, 1977), Vol. 14.