Emergence of a resonance in machine learning

Zheng-Meng Zhai[1] Ling-Wei Kong[1] and Ying-Cheng Lai[1,2,*]
[1] *School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, Arizona 85287, USA*
[2] *Department of Physics, Arizona State University, Tempe, Arizona 85287, USA*

(Received 9 June 2022; revised 1 March 2023; accepted 26 July 2023; published 24 August 2023)

The benefits of noise to applications of nonlinear dynamical systems through mechanisms such as stochastic and coherence resonances have been well documented. Recent years have witnessed a growth of research in exploiting machine learning to predict nonlinear dynamical systems. It has been known that noise can act as a regularizer to improve the training performance of machine learning. Utilizing reservoir computing as a paradigm, we find that injecting noise to the training data can induce a resonance phenomenon with significant benefits to both short-term prediction of the state variables and long-term prediction of the attractor. The optimal noise level leading to the best performance in terms of the prediction accuracy, stability, and horizon can be identified by treating the noise amplitude as one of the hyperparameters for optimization. The resonance phenomenon is demonstrated using two prototypical high-dimensional chaotic systems.

DOI: 10.1103/PhysRevResearch.5.033127

I. INTRODUCTION

A challenging problem in nonlinear dynamics is model-free and data-driven prediction of chaotic systems [1–23]. In general, there are two kinds of forecasting problems: short term and long term. In short-term forecasting, the goal is to predict the detailed dynamical evolution of the state variables from specific initial conditions, typically for a few cycles of oscillation (or Lyapunov times). In long-term prediction, the aim is to generate the attractor of the system with the correct statistical behaviors. According to conventional wisdom, for solving the prediction problems, noise would be detrimental. For example, in short-term prediction, because of the sensitive dependence on initial conditions, noise will make the predicted state evolution diverge exponentially from the true one. In long-term prediction, noise can induce the trajectory to cross the basin boundary, leading to a wrong attractor.

Recent years have witnessed the development of machine-learning techniques for predicting chaotic systems [24–41], where an extensively studied scheme is reservoir computing [42–45]. In machine learning, it has been known that adding noise to the training data can improve the generalizability through the mathematical mechanism of regularization [46]. Quite recently, in a study of reservoir computing used to learn the relationship between different state variables of a chaotic system, it was found that the best performance is achieved when noise amplitudes in the training and testing phases are matched [47]. What is the physical or dynamical mechanism underlying the benefit of noise and how do we find the optimal level of noise?

In this paper, we uncover a resonance phenomenon in which a certain amount of noise can significantly enhance the short-term and long-term prediction accuracy and robustness for chaotic systems, where the optimal noise level can be found through a generalized scheme of hyperparameter optimization. In particular, we consider reservoir computing and inject noise into the input signal. The machine-learning architecture contains a number of hyperparameters and the prediction performance depends on their values. Our simulations reveal that if the hyperparameters are not optimized, noise in the training data can improve to certain extent the prediction performance. However, in order to maximize the predictive power of a reservoir computer, it is necessary to find the optimal values of the hyperparameters, a task that can be accomplished through, e.g., Bayesian optimization [48,49]. The key to identifying the resonance is to treat the noise amplitude as one of the hyperparameters, i.e., to regard it as an intrinsic parameter of the reservoir computer. Bayesian optimization can then yield the optimal noise level. We demonstrate using two prototypical high-dimensional chaotic systems in which noise with the determined amplitude can generate more accurate, robust and stable predictions in both short and long terms. We develop a physical theory by deriving an approximate Langevin equation to understand the emergence of the resonance.

II. RESULTS

The basic principle of reservoir computing and the optimization method are described in Appendix A. There are six hyperparameters to be optimized: the spectral radius ρ of the reservoir network, the scaling factor γ of the input weights, the leakage parameter α, the regularization coefficient β, the link connection probability p of the random network in the

[*] Corresponding author: ylai1@asu.edu

hidden layer, and the noise amplitude σ. To determine the optimal hyperparameter values, we use the SURROGATEOPT function in MATLAB [50], a Bayesian optimization procedure, and employ a surrogate approximation function to estimate the objective function and to find the global minimum through sampling and updating. Specifically, the SURROGATEOPT algorithm [51] first samples several random points and evaluates the objective function at these trial points. The algorithm then creates a surrogate model of the objective function by interpolating a radial basis function through all the random trial points. From the surrogate function, the algorithm identifies the potential minima and samples the points about these minima to update the function.

We demonstrate the benefits of noise to both short-term and long-term prediction using two prototypical chaotic systems: the Mackey-Glass (MG) system described by a nonlinear delay differential equation and the spatiotemporal chaotic Kuramoto-Sivashinsky (KS) system. We use the Bayesian algorithm to obtain the optimal values of the six hyperparameters (including the noise amplitude σ). We then choose a number of σ values away from the optimal value and test the prediction performance. For each fixed σ value, we optimize the other five hyperparameters. For a different value of σ, the set of the other five hyperparameters is then different. As the noise amplitude deviates from the optimal value on either side, there is a gradual deterioration of the prediction performance, signifying the emergence of a resonance.

A. Emergence of a resonance from short-term prediction

Our first example is the MG system [52] described by $\dot{s}(t) = as(t-\tau)/(1+[s(t-\tau)]^c) - bs(t)$, where τ is the time delay, a, b, and c are parameters. The state of the system at time t is determined by the entire prior state history within the time delay, making the phase space of the system infinitely dimensional. To be concrete, we use two values of the time delay: $\tau = 17$ and $\tau = 30$, and fix the other three parameters as $a = 0.2$, $b = 0.1$, and $c = 10$. For $\tau = 17$, the system exhibits a chaotic attractor with one positive Lyapunov exponent: $\lambda_+ \approx 0.006$. For $\tau = 30$, the system has a chaotic attractor with two positive Lyapunov exponents [53]: $\lambda_+ \approx 0.011$ and 0.003. To generate the one-dimensional MG time series data, we integrate the delay differential equation with the time step $h = 0.01$ and generate the training and testing data by sampling the time series every 100 steps: $\Delta t = 100h = 1.0$, where Δt is evolutionary time step of the dynamical network in the hidden layer of the reservoir computer. To remove any transient behavior, we disregard the first $10\,000\Delta t$ in the training data set. The length of training data is $T = 150\,000\Delta t$. The step after the training data marks the start of the testing data, whose length depends on whether the task is to make short-term or long-term prediction. The time series data are preprocessed by using z-score normalization: $z(t) = [s(t) - \bar{s}]/\sigma_s$, where $s(t)$ is the original time series, $\bar{s}$ and σ_s are the mean and standard deviation of $s(t)$, respectively. For $\tau = 17$ and $\tau = 30$ in the MG system, the testing lengths for Bayesian optimization are $T_{\rm opt} = 900\Delta t$ and $300\Delta t$, respectively, which are also the testing lengths for short-term prediction. The so-obtained optimal hyperparameter values are listed in Table I. Figure 1(a) shows, for $\tau = 30$, representative results of short-term prediction of the state evolution, where Gaussian noise with the optimal amplitude is injected into the training time series. Results of long-term prediction in terms of the attractors in the plane $\{X \equiv s(t), Y \equiv s(t-\tau)\}$ are shown in Figs. 1(b) and 1(c). Visually and statistically, the predicted attractor cannot be distinguished from the true attractor. Prediction results for $\tau = 17$ are presented in Fig. 2.

TABLE I. Optimal hyperparameter values for MG and KS.

System	ρ	γ	α	β	p	σ
MG ($\tau = 17$)	1.62	0.55	0.64	$10^{-6.0}$	0.99	$10^{-3.42}$
MG ($\tau = 30$)	1.27	0.23	0.57	$10^{-6.4}$	0.09	$10^{-1.97}$
KS	0.01	0.35	0.62	$10^{-9.0}$	0.21	$10^{-2.35}$

Our second example is the one-dimensional KS system [54,55], a paradigm not only in physics and chemistry but also in applications of reservoir computing for demonstrating the predictive power for high-dimensional dynamical systems [28]. The system equation is

$$\frac{\partial u}{\partial t} + \mu \frac{\partial^4 u}{\partial x^4} + \phi\left(\frac{\partial^2 u}{\partial x^2} + u\frac{\partial u}{\partial x}\right) = 0, \tag{1}$$

where $u(x,t)$ is a scalar field defined in the spatial domain $0 \leqslant x \leqslant L$, μ and ϕ are parameters. We set $\mu = 1$ and $\phi = 1$, and use the periodic boundary condition. As the domain size L increases, the system becomes progressively more high-dimensionally chaotic with the number of Lyapunov

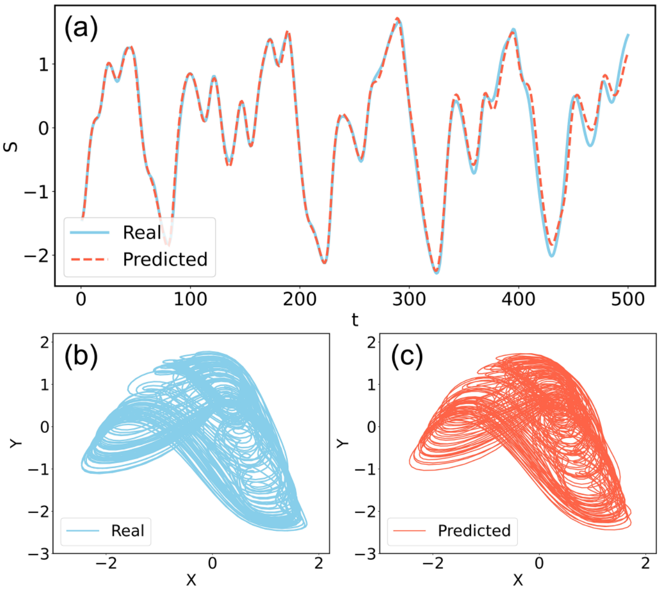

FIG. 1. Short-term and long-term prediction of the MG system for $\tau = 30$. The optimal noise amplitude is $10^{-1.97}$. (a) Machine-predicted system evolution (red trace) in comparison with the ground truth (blue). The predicted state evolution agrees with the true evolution for a time period that contains about 15 local maxima ($T = 500\Delta t$), a result that is significantly better than those without optimal noise. (b), (c) Representation of the true and predicted attractor in the $\{X \equiv s(t), Y \equiv s(t-\tau)\}$ plane. The prediction time length is $T = 10,000\Delta t$.

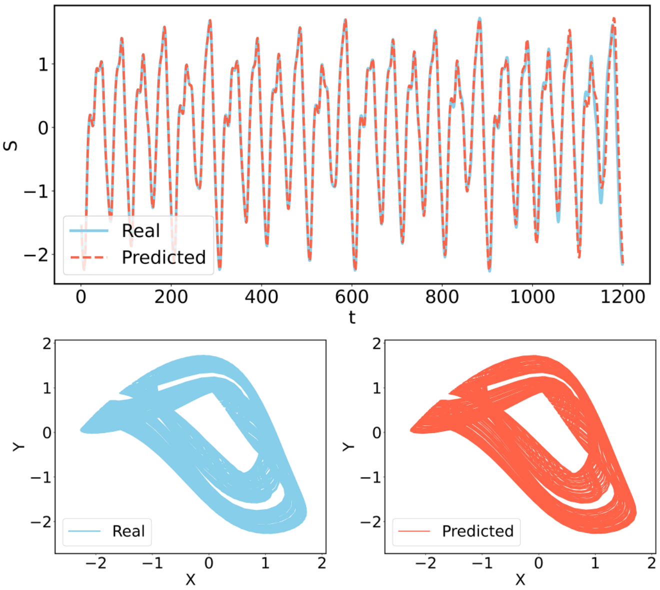

FIG. 2. Short-term and long-term prediction of MG system for $\tau = 17$. The optimal noise amplitude is determined to be $10^{-3.42}$. Top row: Machine-predicted system evolution (red trace) in comparison with the ground truth (blue). The predicted state evolution agrees with the true evolution for more than 20 cycles of oscillation, a result that is significantly better than those without optimal noise. Bottom row: Representation of the true (blue) and the predicted (red) attractor in the $\{X \equiv s(t), Y \equiv s(t-\tau)\}$ plane. The prediction time length is $T = 20,000\Delta t$.

exponents increasing linearly with the system size [56]. As a representative case of high-dimensional chaos, we choose $L = 60$, where the system has seven positive Lyapunov exponents: $\lambda_+ \approx 0.089$, 0.067, 0.055, 0.041, 0.030, 0.005, and 0.003. The length of the training data is about 1000 Lyapunov times (after disregarding a transient of about 300 Lyapunov times), where a Lyapunov time is defined as the inverse of the largest positive exponent. The testing data for short-term and long-term prediction are taken immediately after the training data of six and 100 Lyapunov times, respectively.

Figure 3 shows the results of short-term and long-term predictions of the KS system. It can be seen that the reservoir computing machine with the aid of optimal noise not only can accurately predict the short-term spatiotemporal evolution but also is able to replicate the long-term attractor with the correct statistical behavior. To demonstrate the emergence of a resonance for both short-term and long-term predictions, we ascertain that the optimal noise amplitude values from Bayesian optimization as listed in Table I are indeed optimal. To this end, we vary the noise amplitude (uniformly on a logarithmic scale) in the range $[10^{-8}, 10^{-0.5}]$. For each fixed noise amplitude, we optimize the other five hyperparameters (ρ, γ, α, β, and k). For different values of the noise amplitude, the so-obtained values of the other five hyperparameters are listed in three tables in Appendix B. To characterize the performance of short-term prediction, besides the conventional RMSE, we introduce two additional measures: Prediction horizon and stability, where the former (denoted as t_s) is the maximal time interval during which the RMSE is below a threshold and the latter is the probability that a reservoir computer generates stable dynamical evolution of the target chaotic system in a fixed time window, which is defined as $R_s(r_c) = (1/n)\sum_{i=1}^{n}[\text{RMSE} < r_c]$, where r_c is the RMSE threshold, n is the number of iterations, and $[\cdot] = 1$ if the statement inside is true and zero otherwise.

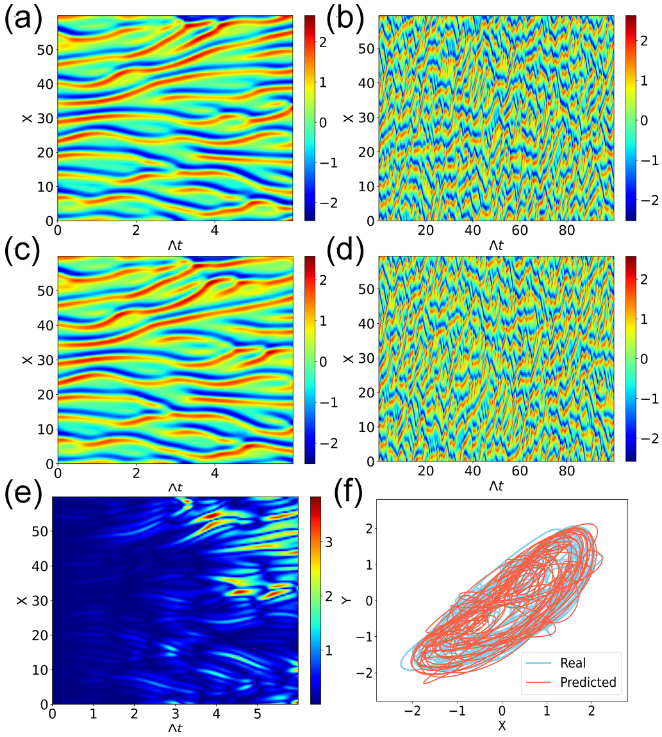

FIG. 3. Short-term and long-term prediction of the KS system. (a), (b) True short-term (six Lyapunov times) and long-term (100 Lyapunov times) spatiotemporal evolution of the nonlinear field $u(x, t)$, respectively, (c), (d) the predicted field $\hat{u}(x, t)$ in short and long terms, respectively. (e) Difference between the predicted and true fields defined as $D(x, t) \equiv \sqrt{[u(x, t) - \hat{u}(x, t)]^2}$. (f) Overlapped image of the true and predicted attractors in terms of the fourth and fifth dimension of the KS system. The values of the optimal hyperparameters (including the optimal noise amplitude) are listed in Table I.

Figure 4 shows the RMSE, the prediction stability $R_s(r_c)$, and the prediction horizon versus the noise amplitude σ for the MG system for $\tau = 30$ (left column, $r_c = 0.1$), as well as the KS system (right column, $r_c = 8.0$). In both cases, an optimal noise level emerges in the sense that a prediction measure versus the noise amplitude exhibits either a bell-shape or an anti-bell-shape type of variation about an optimal point. Figure 4 thus provides strong evidence for a resonance associated with short-term performance of machine-learning prediction of chaotic systems.

Figure 5 illustrates the three quantitative measures (RMSE, prediction stability and horizon) characterizing the short-term prediction versus the noise amplitude for the MG system for $\tau = 17$. The emergence of an optimal noise level can be seen. The results in Figs. 2–5 provide strong evidence for the emergence of an optimal noise level and a resonance in reservoir-computing-based machine learning.

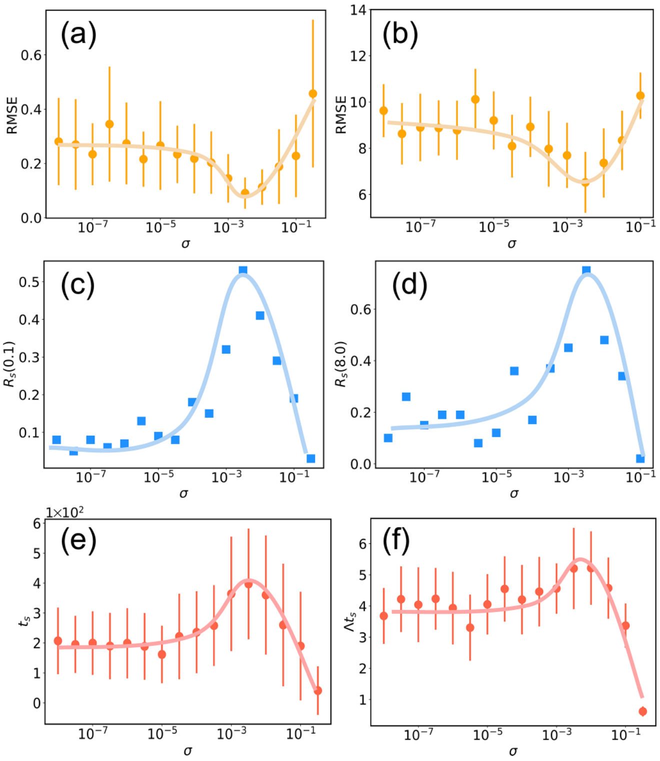

FIG. 4. A resonance associated with short-term prediction of chaotic systems. Shown are three measures of short-term prediction versus the noise amplitude for two examples: left column, MG system for $\tau = 30$ ($r_c = 0.1$, length of prediction time window $= 300\Delta t$); right column, KS system ($r_c = 8.0$, length of prediction time window = five Lyapunov times); top row, RMSE; middle row, prediction stability $R_s(r_c)$; bottom row, prediction horizon t_s. The error bars are obtained from an ensemble of 80 performing reservoir computers. For each chaotic system, a specific and unique noise level emerges at which each prediction measure is optimized, which is characteristic of a resonance.

B. Emergence of a resonance from long-term prediction

We study the beneficial role of noise in long-term prediction of chaotic attractors. Due to the sensitive dependence on initial conditions in chaotic systems, an accurate prediction of the state evolution is possible only within a few Lyapunov times. However, as we have demonstrated in the main text, it is still possible to predict the long-term statistical behavior, e.g., the attractor of the system. If this is the case, the trained reservoir computer has captured the dynamical climate of the target system. It can also occur that a reservoir computer, in spite of training, fails to capture the climate of the target system. In this case, the attractor predicted by the machine deviates from the true one. Remarkably, we find that noise can enhance the reservoir computer's ability to capture the dynamical climate of the target system, providing a beneficial role in long-term prediction.

To compare two attractors, it is necessary to introduce a measure to quantify their mutual deviation. To gain insights, we first generate six examples of long-term prediction of the Mackey-Glass (MG) system: two for $\tau = 17$, two for $\tau = 30$, and long-term prediction of the Kuramoto-Sivashinsky (KS) system. For each example, we generate a case of successful prediction (with noise amplitude as one of the hyperparameters optimized) and a case of unsuccessful prediction (without noise), as shown by the predicted and true chaotic attractors in Figs. 6(a) and 6(b), and in Figs. 7(a) and 7(b). We place a uniform grid in a two-dimensional subspace with cell size $\Delta = 0.05$, count the number of trajectory points in each cell

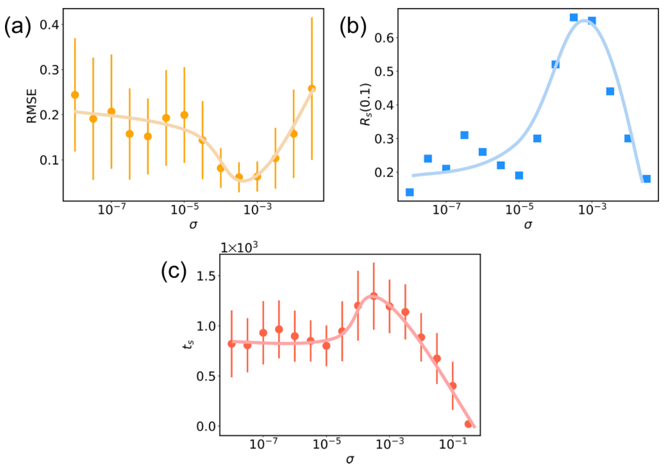

FIG. 5. A resonance associated with short-term prediction for the MG system for $\tau = 17$. Shown are the three measures of short-term prediction versus the noise amplitude: (a) RMSE, (b) prediction stability $R_s(r_c)$, (c) prediction horizon t_s. The error bars are obtained from an ensemble of 80 performing reservoir computers. For this system, a specific and unique noise level emerges at which each prediction measure is optimized, which is characteristic of a resonance. The relevant parameter values are $r_c = 0.1$ and length of prediction time window = 900.

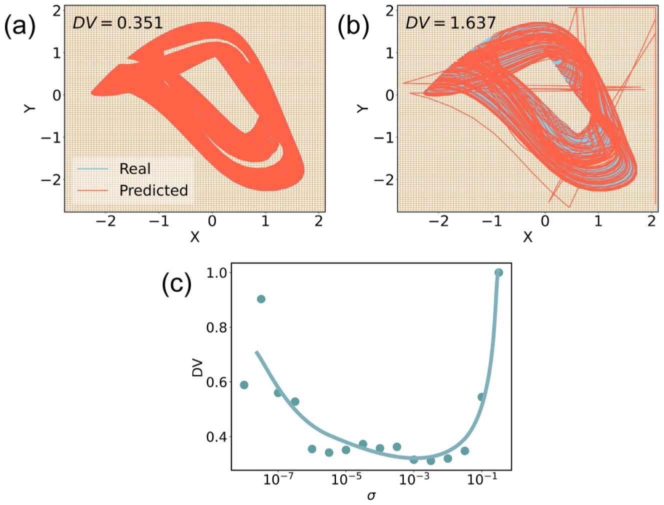

FIG. 6. Quantifying long-term prediction through the deviation value DV. (a) A successful case of attractor prediction for the MG system for $\tau = 17$. (b) An unsuccessful case of attractor prediction for the same system. The two-dimensional phase space is $\{X(t) \equiv s(t), Y(t) \equiv s(t-\tau)\}$. (c) DV versus the noise amplitude for the MG system for $\tau = 17$. There is a range of noise amplitude in which the DV value is minimized, which contains the optimal noise level determined from the corresponding short-term prediction results in Fig. 2.

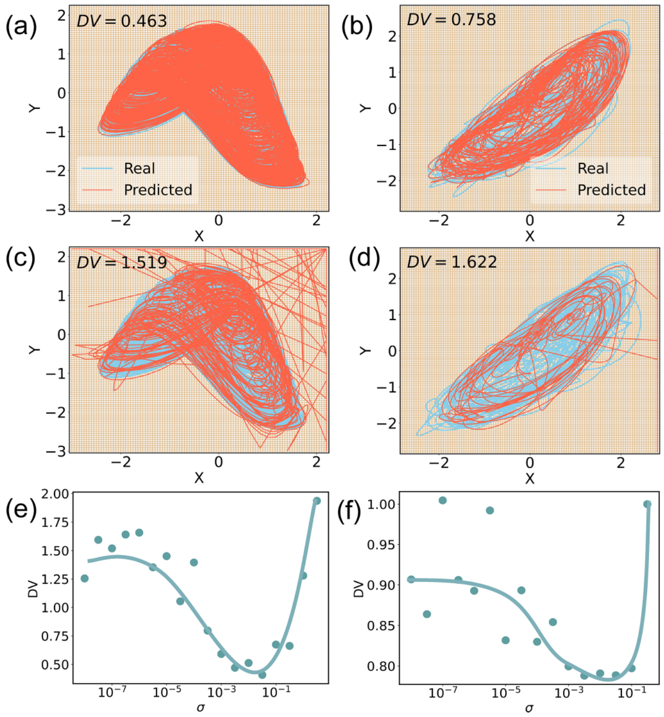

FIG. 7. Quantifying long-term prediction through the deviation value DV. (a), (b) Successful cases of attractor prediction in the presence of optimal noise for the MG system for $\tau = 30$ and KS system, respectively. (c), (d) Unsuccessful cases of attractor prediction without noise for the two systems. The two-dimensional phase space for the MG system is $\{X(t) \equiv s(t), Y(t) \equiv s(t-\tau)\}$. For the KS system, the space is $\{X(t) \equiv u(4,t), Y(t) \equiv u(5,t)\}$. (e), (f) DV versus the noise amplitude for the MG and KS systems, respectively. There exists an optimal noise amplitude at which the DV value is minimized, which agrees with the optimal noise level determined from the corresponding short-term prediction results in Fig. 4.

for both the true and predicted attractors in a fixed time interval, and define the deviation value (DV) as:

$$\mathrm{DV} \equiv \sum_{i=1}^{m_x} \sum_{j=1}^{m_y} \sqrt{(f_{i,j} - \hat{f}_{i,j})^2} \qquad (2)$$

where m_x and m_y are the total numbers of cells in the x and y directions, respectively, $f_{i,j}$ and $\hat{f}_{i,j}$ are the frequencies of visit to the cell (i, j) by the true and predicted trajectories, respectively. If the predicted trajectory leaves the square, we count them as if they belonged to the cells at the boundary where the true trajectory never visits.

The length of the time interval to demonstrate long-term prediction of the MG system for $\tau = 17$ and $\tau = 30$ is $T = 20\,000\Delta t$. For the KS system, the length is 200 Lyapunov times. Different from RMSE, the DV value will not be large even if there is a collapse. It is meaningful to calculate the average DV. We do this using 100 independent realizations of the reservoir computer for each example. Figures 6(a) and 6(b) show that the DV value for the successful case of prediction is much smaller than that for the unsuccessful case, for the MG system for $\tau = 17$. In fact, for the unsuccessful cases without noise, the predictions are so bad that, after a transient time, the predicted trajectories largely deviate from the true attractor. Without noise facilitating the training, such unsuccessful cases of attractor prediction are not uncommon [26]. Figure 6(c) show DV versus the noise amplitude for the MG system for $\tau = 17$. In this case, there is a range of the noise amplitude in which the DV value is minimized. Note that this range contains the optimal value of the noise amplitude from the short-term prediction results in Fig. 2.

Figure 7 illustrates the emergence of a resonance from long-term prediction for the MG system for $\tau = 30$ (left column) and the KS system (right column). In each case, there is an optimal noise amplitude at which the DV value is minimized [Figs. 7(e) and 7(f)], which agrees with the optimal value of the noise amplitude from the short-term prediction results in Fig. 4, providing additional support for the emergence of a resonance in machine learning in terms of long-term prediction of chaotic attractors. Compared with short-term prediction, the resonance associated with long-term prediction is wider about the same optimal noise level. These results provide consistent support for the emergence of a resonance from the perspective of long-term prediction of chaotic attractors.

III. HEURISTIC REASON FOR THE OCCURRENCE OF A RESONANCE

Intuitively, the dynamical mechanism of the resonance is the result of a time-scale match. In particular, the input chaotic signal to the machine has an intrinsic time scale. When noise is present in the input, the recurrent nature of the neural network generates stochastic evolution of the dynamical state, inducing another time scale: the mean first-passage time. When these two time scales match, a resonance emerges. For reservoir computing with nonlinear activation, at the present it is not feasible to develop a quantitative mathematical understanding of the resonance. However, the seminal work by Bollt [40] suggested that an approximate model of linear reservoir computing captures the essential dynamics of the neural learning mechanism. We thus consider this approximate model subject to input noise and provide a heuristic argument that the underlying stochastic dynamics can be described by a Langevin-like equation: $\dot{\mathbf{r}} \approx \mathbf{f}(\mathbf{r}, \mathbf{u}) + \xi$, where $\mathbf{r}$ and $\mathbf{u}$ represent the reservoir hidden state and the input vectors, respectively, the deterministic force $\mathbf{f}(\cdot)$ is a function of $\mathbf{r}$ and $\mathbf{u}$ and ξ is a vector of random fluctuations representing the stochastic perturbations. The time-scale match can be demonstrated by applying the analysis of the Langevin equation in treating noise-enhanced temporal regularity (or coherence resonance) in chaotic systems [57,58].

A resonance in nonlinear dynamical systems is broadly referred to as the phenomenon in which noise can improve the performance of the system. For example, coherence resonance is characterized by the optimization of a measure of the temporal regularity of the state variables by noise, which was originally studied in neural dynamical systems [59] and observed in various other systems, such as climate systems [60], lasers [61], and biological systems [62]. Unlike a stochastic resonance [63–68], which describes the effect of noise on overcoming the system's energy barriers and improving the signal-to-noise ratio, coherence resonance concerns the temporal aspect of the signal and it does not require an external periodic driving.

In general, the underlying mechanism of any resonance phenomenon is a match between two time scales, which occurs at some optimal noise level. For example, in a coupled oscillator system, one time scale can be the characteristic average frequency of the individual oscillators and the other is induced by noise, e.g., the mean first-passage time. We have demonstrated that a certain level of noise in the data can improve, quite remarkably, the ability of a reservoir computer to predict both the short-term dynamical evolution and the long-term invariant sets of chaotic systems. Quantitatively, we find that a number of measures characterizing the short-term and long-term prediction performance exhibit the defining feature of a resonance: there exists an optimal noise amplitude for which the measures are maximized. Because of this remarkable consistency and agreement with the notion of a general resonance in nonlinear systems, we propose that the phenomenon uncovered in our work indeed represents a resonance.

A challenging issue is to identify the underlying dynamical mechanism responsible for the emergence of a resonance in machine learning. It is difficult to apply a mechanical model to the machine-learning system, as the dynamics of the high-dimensional neural network in the hidden layer are extraordinarily complicated. Our approach is to develop an approximate physical picture. Following Bollt's seminal work on explainable reservoir computing [40], we apply stochastic input and derive a Langevin type of equation to obtain a physical understanding of the numerically observed resonance phenomenon.

The state evolution of the recurrent neural network in the hidden layer is described in Appendix A 1. For simplicity, we set $\alpha = 1$ and rewrite the equation of dynamical evolution as

$$\mathbf{r}(t+\Delta t) = \tanh\{\mathcal{A}\cdot\mathbf{r}(t) + \mathcal{W}_{\text{in}}\cdot[\mathbf{u}(t)+\xi(t)]\}, \tag{3}$$

where Δt is the time step, $\xi(t)$ represents the noise added to the input signal, and the activation is described by the hyperbolic tangent function $\tanh(x)$. For $x \ll 1$, we have $\tanh(x) \sim x$ and obtain the special class of linear reservoir computers [40]:

$$\mathbf{r}(t+\Delta t) \approx \mathcal{A}\cdot\mathbf{r}(t) + \mathcal{W}_{\text{in}}\cdot\mathbf{u}(t) + \mathcal{W}_{\text{in}}\cdot\xi(t). \tag{4}$$

To simplify notation, we set $\Delta t = 1$. Successive iterations of Eq. (4) are

$$\begin{aligned}
\mathbf{r}(2) &= \mathcal{A}\cdot\mathbf{r}(1) + \mathcal{W}_{\text{in}}\cdot[\mathbf{u}(1)+\xi(1)] = \mathcal{W}_{\text{in}}\cdot[\mathbf{u}(1)+\xi(1)],\\
\mathbf{r}(3) &= \mathcal{A}\cdot\mathbf{r}(2) + \mathcal{W}_{\text{in}}\cdot[\mathbf{u}(2)+\xi(2)] = \mathcal{A}\cdot\mathcal{W}_{\text{in}}[\mathbf{u}(1)+\xi(1)] + \mathcal{W}_{\text{in}}\cdot[\mathbf{u}(2)+\xi(2)]\\
&\vdots\\
\mathbf{r}(t+1) &= \mathcal{A}\cdot\mathbf{r}(t) + \mathcal{W}_{\text{in}}\cdot[\mathbf{u}(t)+\xi(t)]\\
&= \mathcal{A}\cdot\{\mathcal{A}\cdot\mathbf{r}(t-1) + \mathcal{W}_{\text{in}}\cdot[\mathbf{u}(t-1)+\xi(t-1)]\} + \mathcal{W}_{\text{in}}\cdot[\mathbf{u}(t)+\xi(t)]\\
&\vdots\\
&= \mathcal{A}^{k-1}\cdot\mathcal{W}_{\text{in}}\cdot[\mathbf{u}(1)+\xi(1)] + \mathcal{A}^{k-2}\cdot\mathcal{W}_{\text{in}}\cdot[\mathbf{u}(2)+\xi(2)] + \dots\\
&\quad + \mathcal{A}\cdot\mathcal{W}_{\text{in}}\cdot[\mathbf{u}(t-1)+\xi(t-1)] + \mathcal{W}_{\text{in}}\cdot[\mathbf{u}(t)+\xi(t)]\\
&= \sum_{i=1}^{t}\mathcal{A}^{i-1}\cdot\mathcal{W}_{\text{in}}\cdot[\mathbf{u}(t-i+1)+\xi(t-i+1)],
\end{aligned} \tag{5}$$

where $\mathcal{A}^0 = \mathcal{I}$. Since the output matrix $\mathcal{W}_{\text{out}}$ maps $\mathbf{r}(t)$ into the output signal $\mathbf{v}(t)$ that has the same dimension as that of the input vector $\mathbf{u}(t)$, we have

$$\mathbf{v}(t+1) = \mathcal{W}_{\text{out}}\cdot\mathbf{r}(t+1) = \sum_{i=1}^{t}\mathcal{W}_{\text{out}}\cdot\mathcal{A}^{i-1}\cdot\mathcal{W}_{\text{in}}\cdot[\mathbf{u}(t-i+1)+\xi(t-i+1)], \tag{6}$$

indicating that a linear reservoir computer yields a vector autoregressive process (VAR) with a general form [40]:

$$\begin{aligned}
\mathbf{y}(t+1) = c + A_1\cdot\mathbf{x}(t) + A_2\cdot\mathbf{x}(t-1) + \dots\\
+ A_t\mathbf{x}(1) + \xi(t+1),
\end{aligned} \tag{7}$$

where $\mathbf{x}$ and $\mathbf{y}$ represent the input and output, respectively, c is a constant term, and $A_1, A_2, \dots, A_t$ are coefficient matrices, and ξ denotes the stochastic process.

Note that the reservoir state evolution Eq. (4) and its recursion Eq. (5) can be cast into the form

$$\frac{d\mathbf{r}}{dt} = [\mathcal{A} - \mathcal{I}]\cdot\mathbf{r} + \mathcal{W}_{\text{in}}\cdot\mathbf{u} + \mathcal{W}_{\text{in}}\cdot\xi, \tag{8}$$

which is similar in mathematical form to the Langevin equation:

$$\frac{dx}{dt} = f(x) + g(x)\xi(t) \tag{9}$$

that describes a particle moving under the influence of two forces: a deterministic force $f(x_l)$ and a stochastic force

$g(x_l)\xi(t)$. Comparing Eqs. (8) and (9), we have that $[\mathcal{A}-\mathcal{I}]\cdot\mathbf{r}+\mathcal{W}_{\rm in}\cdot\mathbf{u}$ represents the deterministic force while $\mathcal{W}_{\rm in}\cdot\xi$ is the stochastic force that provides the random driving to the reservoir intrinsic state. The Langevin equation of the form Eq. (4) was shown previously through the corresponding Fokker-Planck equation to yield a stochastic time scale required for matching with the dynamical time scale of the system in the context of coherence resonance [57,58]. The approximate equivalence of Eq. (4) to the particular form of the Langevin equation provides a physical reason for a resonance to arise in reservoir computing, as we have demonstrated numerically.

IV. DISCUSSION

To summarize, we have uncovered the emergence of a resonance in machine-learning prediction of chaotic systems. Focusing on reservoir computing, we find that injecting noise into the training data can be beneficial to both short- and long-term predictions. In particular, for short-term prediction, a number of characterizing quantities such as the prediction accuracy, stability, and horizon can be maximized by an optimal level of noise that can be found through hyperparameter optimization. For long-term prediction, optimal noise can significantly increase the chance for the machine generated trajectory to stay in the vicinity of (or to shadow) the true attractor of the target chaotic system. Intuitively, training with noise can enhance the machine's tolerance to chaotic fluctuations, which can be beneficial for the machine to learn the dynamical climate of the target chaotic system. This suggests that the optimal noise level should be on the same order of magnitude as the one-step prediction error in noiseless prediction, which is indeed so as verified by our numerical examples. Pertinent issues such as the requirement of prediction time for the emergence of a resonance, robustness of resonance against different scenarios of noise injection, and the beneficial role of noise in reducing the reservoir network size and computational complexity are addressed in Appendix C. Our work extends the ubiquitous phenomena of stochastic [63–68] and coherence [57,59] resonances in nonlinear dynamical systems to the realm of machine learning, where deliberate noise combined with hyperparameter optimization can be a practically feasible approach to enhancing the predictive power.

We note that, previously the role of noise in neural network training was studied, e.g., adding noise to the training data for convolutional neural networks can play the role of regularization to reduce overfitting in the learning models [69]. In reinforcement learning, injecting noise into the signals can help the system reach the persistent excitation condition to facilitate parameter estimation [70,71]. How noise negatively affects the prediction of chaotic systems has recently been considered [72], where long short-term memory machines tend to be more resistant to noise than other machine-learning methods. The beneficial role of noise in machine-learning prediction has also been recognized [46,73–75]. We present a systematic study of the interplay between noise and machine-learning prediction of dynamical systems in this work, along with the demonstration of the resonance phenomenon in machine learning.

All relevant data are available from the authors upon request. All relevant computer codes are available from the authors upon request.

ACKNOWLEDGMENT

This work was supported by Air Force Office of Scientific Research under Grant No. FA9550-21-1-0438.

APPENDIX A: RESERVOIR COMPUTING WITH INPUT NOISE FOR PREDICTING DYNAMICAL SYSTEMS

1. Principle of reservoir computing

A reservoir computer is essentially a recurrent neural network (RNN), which consists of three components: an input layer, a hidden layer, and an output layer. Compared with the conventional RNNs, the key advantage of reservoir computing lies in its computational efficiency: the input weights and the hidden layer neural network are predefined and only the weights of the output layer need to be determined from training through a standard linear regression.

As illustrated in Fig. 8, the input matrix $\mathcal{W}_{\rm in}$ maps the input signal $\mathbf{u}(t)$ into a hidden layer. The hidden, recurrent layer hosts a neural network characterized by the adjacency matrix $\mathcal{A}$, whose state vector is $\mathbf{r}(t)$, where the ith entry represents the dynamical state of the ith neuron in the network. The dynamical evolution of $\mathbf{r}(t)$ is determined by both the input and the recurrent structure:

$$\mathbf{r}(t+\Delta t)=(1-\alpha)\mathbf{r}(t)+\alpha\cdot\tanh\left[\mathcal{A}\cdot\mathbf{r}(t)+\mathcal{W}_{\rm in}\cdot\mathbf{u}(t)\right], \tag{A1}$$

where α is the leakage parameter determining the temporal scale of the neural network, Δt is the time step of the dynamical evolution, and the activation is described by the hyperbolic tangent function $\tanh(x)$. The output matrix $\mathcal{W}_{\rm out}$ maps $\mathbf{r}(t)$ into the output signal $\mathbf{v}(t)$ that typically has the same dimension as the input vector $\mathbf{u}(t)$.

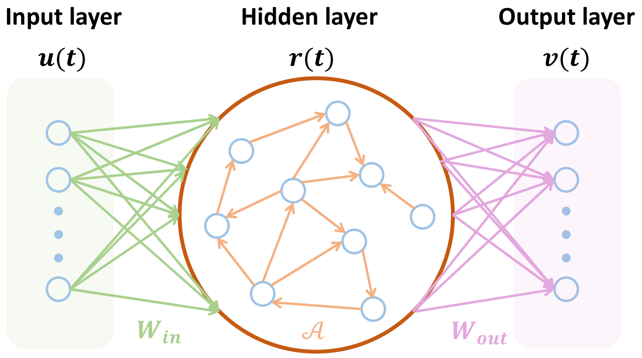

FIG. 8. Reservoir computing structure. A reservoir computer consists of three components: an input layer, a hidden layer, and an output layer. The vectors $\mathbf{u}(t)$, $\mathbf{r}(t)$, and $\mathbf{v}(t)$ are the input signal, the dynamical state of the network in the hidden layer, and the output signal, respectively. The matrices $\mathcal{W}_{\rm in}$, $\mathcal{A}$, and $\mathcal{W}_{\rm out}$ represent the input weights, the network structure, and the output weights, respectively. The elements of $\mathcal{W}_{\rm in}$ and $\mathcal{A}$ are predefined and fixed. The matrix $\mathcal{W}_{\rm out}$ is determined by training through a linear regression.

Let D_{in} and D_r be the dimension of the input vector $\mathbf{u}(t)$ and of the hidden-layer state vector $\mathbf{r}(t)$, respectively. The matrix $\mathcal{W}_{\text{in}}$ thus has the dimension $D_r \times D_{\text{in}}$, where $D_r \gg D_{\text{in}}$ so that $\mathcal{W}_{in}$ maps a low-dimensional input vector to a high-dimensional hidden state vector. Prior to training, the weights (elements) of $\mathcal{W}_{\text{in}}$ are chosen uniformly from the interval $[-\gamma, \gamma]$. The dimension of the adjacency matrix $\mathcal{A}$ is $D_r \times D_r$, which characterizes a symmetric random network with link probability p. The nonzero elements of $\mathcal{A}$ are drawn from a Gaussian normal distribution of zero mean and unit variance. We rescale $\mathcal{A}$ so that its spectral radius is given by the hyperparameter ρ. The output matrix $\mathcal{W}_{\text{out}}$ has the dimension $D_{\text{out}} \times D_r$, which is determined by l-2 linear regression (ridge regression) as

$$\mathcal{W}_{\text{out}} = \mathcal{U} \cdot \mathcal{R}'^{\intercal}(\mathcal{R}' \cdot \mathcal{R}'^{\intercal} + \beta\mathcal{I})^{-1}, \tag{A2}$$

where $\mathcal{I}$ is the identity matrix of dimension D_r, β is the l-2 regularization coefficient, $\mathcal{U}$ and $\mathcal{R}'$ consist of $\mathbf{u}(t)$ and $\mathbf{r}'(t)$ at all time steps, respectively, in which a column represents the corresponding vector at a specific time step. The vector $\mathbf{r}'(t)$ is identical to $\mathbf{r}(t)$ except that all the entries in the even rows are squared. Note that $\mathbf{u}(t)$ is the training target for the reservoir to produce a one-step prediction.

We inject Gaussian white noise of zero mean and standard deviation σ to each dimension of the training data and investigate the prediction performance for different values of the noise amplitude σ. We treat σ as one of the hyperparameters of the whole reservoir computer. For prediction and validation, no noise is applied. In particular, during the prediction phase, the output vector $\mathbf{v}(t)$ becomes the input vector $\mathbf{u}(t)$ and the reservoir computer generates one-step predictions. A step-by-step iterative process leads to a prediction signal, whose accuracy is determined by the real, noiseless testing data. For validation, we measure the performance of the trained reservoir computer using the root mean-square error (RMSE):

$$\text{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{T_{\text{stp}}}\frac{1}{N}\sum_{t=1}^{T_{\text{stp}}}\sum_{n=1}^{N}[y_n(t) - \hat{y}_n(t)]^2}, \tag{A3}$$

where $\mathbf{y}$ and $\hat{\mathbf{y}}$ are the real and predicted signals, respectively, $y_n(t)$ represent the nth component of $\mathbf{y}$ at time step t, and T_{stp} is the prediction time. We use the RMSE to characterize the short-term prediction performance, typically for about 4∼5 Lyapunov times.

2. Implementation of reservoir computing and hyperparameter optimization

In the past few years, reservoir computing has been shown to be effective for modeling the dynamics of low- and high-dimensional chaotic systems [24–41]. As denoted in Ref. [35], if a reservoir computer can acquire the full state dynamics in the training phase, it outperforms the back propagation through time approaches, such as long short-term memory (LSTM) systems with respect to both short-term and long-term prediction even with much less training time. A main advantage of reservoir computing is that the input weights and the hidden-layer neural network are predefined prior to training, and only the output parameters need to be optimized at the end of training phase through, e.g., an l-2 linear regression (ridge regression).

We choose MATLAB so that we can readily build up the reservoir-computing framework through adjusting the network structure, parameters, and noise input setting. In fact, a number of reservoir-computing packages now exist, such as RESERVOIRPY and ECHOTORCH in PYTHON or RESERVOIRCOMPUTING in JULIA. Regardless of the programming languages, the core algorithm of reservoir computing is the same and the implementation of the algorithms are quite similar.

It is essential to choose an appropriate optimization algorithm to search for the optimal hyperparameters. In our work, we used Bayesian optimization to determine the optimal hyperparameter values. This is because some traditional optimization algorithms, such as the gradient-free optimization algorithms that use grid or random search, may not be suitable for solving such complex problems as predicting chaotic systems, while Bayesian optimization has the ability to handle difficult problems with fewer iterations [54]. The Bayesian optimization method can be implemented using PYTHON or other languages. Different packages for Bayesian optimization are now available, such as BAYESIAN-OPTIMIZATION and BOTORCH in PYTHON.

APPENDIX B: BAYESIAN OPTIMAL HYPERPARAMETER VALUES

Tables II, III, and IV list the Bayesian optimal hyperparameter values for a set of systematically varying noise amplitudes to demonstrate the emergence of a resonance for the three examples in the main text.

TABLE II. Optimal hyperparameter values for MG system with $\tau = 17$.

noise amplitude σ	ρ	b	α	β	p
$10^{-8.0}$	1.38	0.88	0.45	$10^{-3.5}$	0.58
$10^{-7.5}$	1.27	0.64	0.31	$10^{-5.2}$	0.53
$10^{-7.0}$	1.31	0.62	0.30	$10^{-4.2}$	0.99
$10^{-6.5}$	1.18	0.49	0.32	$10^{-5.6}$	0.48
$10^{-6.0}$	1.17	0.56	0.30	$10^{-4.5}$	0.77
$10^{-5.5}$	1.32	0.20	0.43	$10^{-5.1}$	0.55
$10^{-5.0}$	1.30	0.25	0.51	$10^{-4.7}$	0.54
$10^{-4.5}$	1.23	0.57	0.40	$10^{-4.2}$	0.70
$10^{-4.0}$	1.31	0.54	0.84	$10^{-5.3}$	0.25
$10^{-3.5}$	1.38	0.71	0.34	$10^{-5.8}$	0.33
$10^{-3.0}$	1.68	0.71	0.55	$10^{-7.4}$	0.85
$10^{-2.5}$	1.48	0.63	0.36	$10^{-6.7}$	0.71
$10^{-2.0}$	1.92	1.73	0.26	$10^{-5.6}$	0.90
$10^{-1.5}$	1.60	1.15	0.49	$10^{-6.3}$	0.95
$10^{-1.0}$	1.38	0.76	0.48	$10^{-4.4}$	0.72
$10^{-0.5}$	0.88	0.19	0.96	$10^{-4.6}$	0.84

TABLE III. Optimal hyperparameter values for MG system with $\tau = 30$.

noise amplitude σ	ρ	b	α	β	p
$10^{-8.0}$	1.55	0.12	0.60	$10^{-6.0}$	0.09
$10^{-7.5}$	1.59	0.22	0.22	$10^{-7.1}$	0.81
$10^{-7.0}$	1.29	0.19	0.17	$10^{-7.5}$	0.34
$10^{-6.5}$	2.08	0.11	0.50	$10^{-7.2}$	0.17
$10^{-6.0}$	1.35	0.25	0.18	$10^{-7.1}$	0.12
$10^{-5.5}$	1.28	0.14	0.74	$10^{-4.6}$	0.24
$10^{-5.0}$	1.45	0.15	0.94	$10^{-4.6}$	0.19
$10^{-4.5}$	1.51	0.11	0.71	$10^{-5.8}$	0.34
$10^{-4.0}$	1.63	0.15	0.59	$10^{-6.3}$	0.39
$10^{-3.5}$	1.53	0.10	0.69	$10^{-7.5}$	0.97
$10^{-3.0}$	1.09	0.21	0.66	$10^{-6.1}$	0.32
$10^{-2.5}$	1.14	0.15	0.71	$10^{-7.4}$	0.95
$10^{-2.0}$	2.00	0.24	0.41	$10^{-7.2}$	0.17
$10^{-1.5}$	0.87	0.72	0.20	$10^{-7.5}$	0.37
$10^{-1.0}$	1.72	0.31	0.36	$10^{-7.5}$	0.15
$10^{-0.5}$	1.57	0.19	0.41	$10^{-4.7}$	0.95
$10^{+0.0}$	1.52	0.07	0.63	$10^{-3.8}$	0.30
$10^{+0.5}$	2.46	0.01	0.35	$10^{-6.5}$	0.56

APPENDIX C: PERTINENT ISSUES

1. Requirement of prediction time for the emergence of a resonance

The emergence of a resonance in predicting the dynamical evolution of the chaotic system depends on the prediction time T_{stp} in the definition of RMSE. For a short prediction time, the RMSE is generally small, regardless of the training noise amplitude. As T_{stp} increases, the benefits of noise begin to stand out, leading to the emergence of a resonance. This behavior is illustrated in Fig. 9 for the MG and KS systems, where the color-coded RMSE values in the parameter plane $(T_{\mathrm{stp}}, \sigma)$ are shown. Here, for better visualization, the RMSE values are normalized to the unit interval with respect to the variation in the noise amplitude for each fixed value of T_{stp}.

TABLE IV. Optimal hyperparameter values for KS system.

noise amplitude σ	ρ	b	α	β	p
$10^{-8.0}$	0.01	0.43	0.83	$10^{-9.0}$	0.14
$10^{-7.5}$	0.01	0.27	0.82	$10^{-7.5}$	0.36
$10^{-7.0}$	0.01	0.31	0.76	$10^{-7.4}$	0.49
$10^{-6.5}$	0.01	0.33	0.90	$10^{-8.4}$	0.13
$10^{-6.0}$	0.01	0.31	0.87	$10^{-6.4}$	0.37
$10^{-5.5}$	0.01	0.43	0.78	$10^{-8.0}$	0.59
$10^{-5.0}$	0.01	0.29	0.82	$10^{-5.7}$	0.71
$10^{-4.5}$	0.01	0.27	0.91	$10^{-7.2}$	0.35
$10^{-4.0}$	0.01	0.26	0.81	$10^{-6.5}$	0.48
$10^{-3.5}$	0.01	0.25	0.81	$10^{-6.8}$	0.13
$10^{-3.0}$	0.01	0.29	0.73	$10^{-6.9}$	0.86
$10^{-2.5}$	0.01	0.08	0.67	$10^{-9.0}$	0.09
$10^{-2.0}$	0.01	0.15	0.65	$10^{-8.3}$	0.32
$10^{-1.5}$	0.01	0.09	0.95	$10^{-8.4}$	0.51
$10^{-1.0}$	0.01	0.18	0.37	$10^{-9.0}$	0.34
$10^{-0.5}$	0.16	0.00	0.58	$10^{-5.4}$	0.26

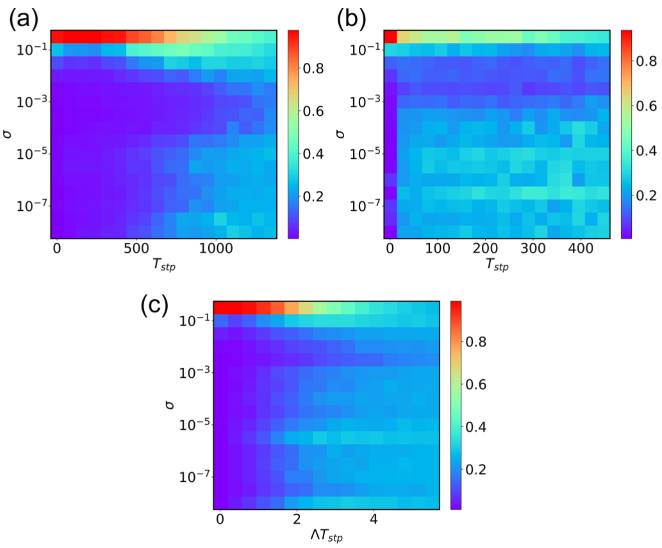

FIG. 9. Emergence of a resonance. Shown are the color-coded normalized RMSE values in the parameter plane $(T_{\mathrm{stp}}, \sigma)$ for (a), (b) MG system for $\tau = 17$ and $\tau = 30$, respectively, and (c) KS system. To reduce the statistical fluctuations, the normalized RMSE values are calculated from an ensemble of 80 independently trained reservoir computers.

As shown in Fig. 9(a), for the MG system for $\tau = 17$, a resonance emerges for $T_{\mathrm{stp}} \gtrsim 600\Delta t$, whereas for $\tau = 30$, a resonance emerges almost immediately as T_{stp} increases from zero, as shown in Fig. 9(b). For the KS system, it takes about two Lyapunov times for a resonance to emerge, as shown in Fig. 9(c).

2. Robustness of resonance against different scenarios of noise injection

The results in the main text and discussed so far here are under the scenario where noise is injected into the entire training set. Specifically, we generate a matrix of Gaussian noise of zero mean and standard deviation σ, which has the same dimension and time length as training data set, and add this noise matrix directly to the normalized training data matrix. When updating the reservoir state with the input training data as a driving force, noise appears directly at the input. Simultaneously, the same noise is present at the output layer during the linear regression.

What if noise is added only to the input layer without appearing in the regression step? Figure 10 shows the RMSE versus the noise amplitude for the three examples in Fig. 9, together with the results from the scenario where noise is applied at both the input and output layers (for comparison). A resonance arises under both noise-injection scenarios. In fact, there is little difference between the results from the two scenarios, indicating that the occurrence of the resonance is robust with respect to the ways by which noise is supplied to the reservoir computer.

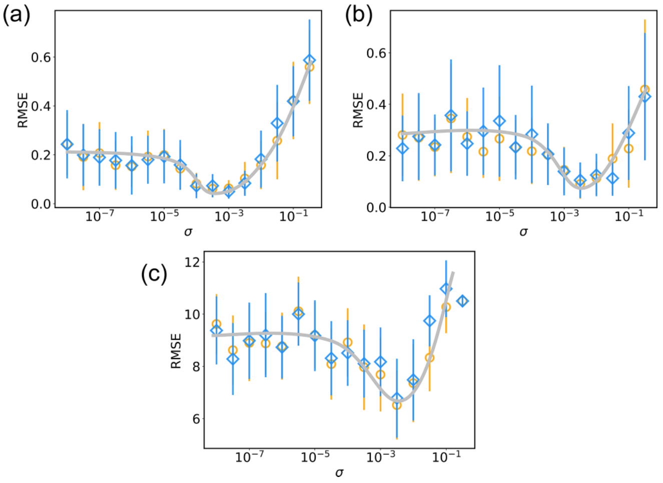

FIG. 10. Resonance under two different noise-inputting scenarios. The two scenarios are: adding noise to both the input and output layers (yellow circles) and injecting noise to the input layer only (blue diamonds) for (a), (b) MG system for $\tau = 17$ and $\tau = 30$, respectively, and (c) KS system. The error bars are obtained using an ensemble of 80 independent realizations of the reservoir computer. Each data point is the result of the ensemble average of 80 best results out of 100 independent realizations of the reservoir computer.

3. Beneficial role of noise in reducing the reservoir network size and computational complexity

In general, the predictive power of a reservoir computer can be improved by increasing the size D_r of the random network in the hidden layer to enable the neural machine to generate more complex and richer dynamics. However, increasing the network size leads to higher computational complexity. Can noise be used to reduce the network size while maintaining the prediction accuracy?

Figure 11 shows, for the three examples in Figs. 9 and 10, the RMSE of short-term prediction versus the network size for four different values of the noise amplitude. For each example, the blue points are for a noise amplitude close to the optimal value for the resonance. It can be seen that optimal noise can lead to a tremendous reduction in the network size. For example, for the MG system for $\tau = 17$ in Fig. 11(a), when optimal noise is added to the training data, the RMSE becomes small as the network size exceeds about 600, whereas this low value of RMSE can never be achieved for near-zero noise (e.g., $\sigma = 10^{-8}$) even if the network size is increased to 3000. A similar behavior occurs for the other two examples, as shown in Figs. 11(b) and 11(c).

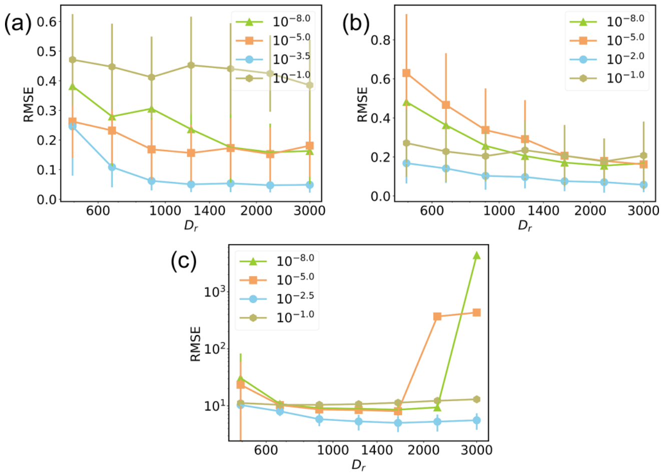

FIG. 11. Demonstration of the beneficial role of noise in reducing the network size (thus computational complexity) while maintaining the prediction accuracy. Shown is the RMSE for short-term prediction versus the size D_r of the hidden-layer network for four different values of the noise amplitude for (a), (b) MG system for $\tau = 17$ and $\tau = 30$, respectively, and (c) KS system. The blue circles correspond to the case of optimal noise level at which a resonance arises, for which the RMSE values are low even for small network size. For different network sizes and noise amplitudes, the values of the five hyperparameters are fixed, which is the reason for the abnormal increase in RMSE at large network size in (c) as predicting the dynamical evolution of the KS system depends sensitively on the hyperparameters. Overall, with optimal noise, the reservoir computer can achieve a high prediction accuracy that cannot be achieved even with much larger networks without noise or when the noise level is not optimal.

[1] M. Casdagli, Nonlinear prediction of chaotic time series, Physica D **35**, 335 (1989).

[2] G. Sugihara, B. T. Grenfell, and R. M. May, Distinguishing error from chaos in ecological time series, Philos. Trans. R. Soc. London B **330**, 235 (1990).

[3] J. Kurths and A. A. Ruzmaikin, On forecasting the sunspot numbers, Sol. Phys. **126**, 407 (1990).

[4] P. Grassberger and T. Schreiber, Nonlinear time sequence analysis, Int. J. Bifurcation Chaos **01**, 521 (1991).

[5] G. Gouesbet, Reconstruction of standard and inverse vector fields equivalent to a Rössler system, Phys. Rev. A **44**, 6264 (1991).

[6] A. A. Tsonis and J. B. Elsner, Nonlinear prediction as a way of distinguishing chaos from random fractal sequences, Nature (London) **358**, 217 (1992).

[7] E. Baake, M. Baake, H. G. Bock, and K. M. Briggs, Fitting ordinary differential equations to chaotic data, Phys. Rev. A **45**, 5524 (1992).

[8] J. P. Crutchfield and B. S. McNamara, Equations of motion from a data series, Complex Sys. **1**, 417 (1987).

[9] T. Sauer, Reconstruction of Dynamical Systems from Interspike Intervals, Phys. Rev. Lett. **72**, 3811 (1994).

[10] G. Sugihara, Nonlinear forecasting for the classification of natural time series, Philos. Trans. R. Soc. London A **348**, 477 (1994).

[11] U. Parlitz, Estimating Model Parameters from Time Series by Autosynchronization, Phys. Rev. Lett. **76**, 1232 (1996).

[12] S. J. Schiff, P. So, T. Chang, R. E. Burke, and T. Sauer, Detecting dynamical interdependence and generalized synchrony

through mutual prediction in a neural ensemble, Phys. Rev. E **54**, 6708 (1996).

[13] R. Hegger, H. Kantz, and T. Schreiber, Practical implementation of nonlinear time series methods: The TISEAN package, Chaos **9**, 413 (1999).

[14] E. M. Bollt, Controlling chaos and the inverse frobenius-perron problem: global stabilization of arbitrary invariant measures, Int. J. Bifurcation Chaos **10**, 1033 (2000).

[15] R. Hegger, H. Kantz, L. Matassini, and T. Schreiber, Coping with Nonstationarity by Overembedding, Phys. Rev. Lett. **84**, 4092 (2000).

[16] S. Sello, Solar cycle forecasting: a nonlinear dynamics approach, Astron. Astrophys. **377**, 312 (2001).

[17] T. Matsumoto, Y. Nakajima, M. Saito, J. Sugi, and H. Hamagishi, Reconstructions and predictions of nonlinear dynamical systems: a hierarchical bayesian approach, IEEE Trans. Signal Proc. **49**, 2138 (2001).

[18] L. A. Smith, What might we learn from climate forecasts? Proc. Natl. Acad. Sci. USA **99**, 2487 (2002).

[19] K. Judd, Nonlinear state estimation, indistinguishable states, and the extended kalman filter, Physica D **183**, 273 (2003).

[20] T. D. Sauer, Reconstruction of Shared Nonlinear Dynamics in a Network, Phys. Rev. Lett. **93**, 198701 (2004).

[21] C. Tao, Y. Zhang, and J. J. Jiang, Estimating system parameters from chaotic time series with synchronization optimized by a genetic algorithm, Phys. Rev. E **76**, 016209 (2007).

[22] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi, Predicting Catastrophes in Nonlinear Dynamical Systems by Compressive Sensing, Phys. Rev. Lett. **106**, 154101 (2011).

[23] W.-X. Wang, Y.-C. Lai, and C. Grebogi, Data based identification and prediction of nonlinear and complex dynamical systems, Phys. Rep. **644**, 1 (2016).

[24] N. 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(R) (2015).

[25] L. 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).

[26] J. Pathak, Z. Lu, B. Hunt, M. Girvan, and E. Ott, Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data, Chaos **27**, 121102 (2017).

[27] T. Duriez, S. L. Brunton, and B. R. Noack, *Machine Learning Control-Taming Nonlinear Dynamics and Turbulence* (Springer, Berlin, 2017).

[28] J. 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).

[29] T. L. Carroll, Using reservoir computers to distinguish chaotic signals, Phys. Rev. E **98**, 052209 (2018).

[30] Z. S. Roland and U. Parlitz, Observing spatio-temporal dynamics of excitable media using reservoir computing, Chaos **28**, 043118 (2018).

[31] A. Griffith, A. Pomerance, and D. J. Gauthier, Forecasting chaotic systems with very low connectivity reservoir computers, Chaos **29**, 123108 (2019).

[32] J. Jiang and Y.-C. Lai, Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks: Role of network spectral radius, Phys. Rev. Res. **1**, 033056 (2019).

[33] H. Fan, J. Jiang, C. Zhang, X. Wang, and Y.-C. Lai, Long-term prediction of chaotic systems with machine learning, Phys. Rev. Res. **2**, 012080(R) (2020).

[34] C. Klos, Y. F. Kalle Kossio, S. Goedeke, A. Gilra, and R.-M. Memmesheimer, Dynamical Learning of Dynamics, Phys. Rev. Lett. **125**, 088103 (2020).

[35] P. R. Vlachasa, J. Pathak, B. R. Hunt, T. P. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos, Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics, Neural Networks **126**, 191 (2020).

[36] L.-W. Kong, H.-W. Fan, C. Grebogi, and Y.-C. Lai, Machine learning prediction of critical transition and system collapse, Phys. Rev. Res. **3**, 013090 (2021).

[37] J. Z. Kim, Z. Lu, E. Nozari, G. J. Pappas, and D. S. Bassett, Teaching recurrent neural networks to infer global temporal structure from local examples, Nature Mach. Intell. **3**, 316 (2021).

[38] H. Fan, L.-W. Kong, Y.-C. Lai, and X. Wang, Anticipating synchronization with machine learning, Phys. Rev. Res. **3**, 023237 (2021).

[39] L.-W. Kong, H.-W. Fan, C. Grebogi, and Y.-C. Lai, Emergence of transient chaos and intermittency in machine learning, J. Phys. Complex. **2**, 035014 (2021).

[40] E. Bollt, On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD, Chaos **31**, 013108 (2021).

[41] D. J. Gauthier, E. Bollt, A. Griffith, and W. A. S. Barbosa, Next generation reservoir computing, Nature Commun. **12**, 5564 (2021).

[42] 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**, 13 (2001).

[43] W. Maass, T. Nachtschlaeger, and H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Comput. **14**, 2531 (2002).

[44] H. Jaeger and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science **304**, 78 (2004).

[45] G. Manjunath and H. Jaeger, Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks, Neural Comput. **25**, 671 (2013).

[46] C. M. Bishop, Training with noise is equivalent to tikhonov regularization, Neural Comput. **7**, 108 (1995).

[47] C. Nathe, C. Pappu, N. A. Mecholsky, J. Hart, T. Carroll, and F. Sorrentino, Reservoir computing with noise, Chaos **33**, 041101 (2023).

[48] J. Snoek, H. Larochelle, and R. P. Adams, Practical Bayesian optimization of machine learning algorithms, in *Advances in Neural Information Processing Systems*, edited by F. Pereira, C. Burges, L. Bottou, and K. Weinberger (Curran Associates, Inc., Brooklyn, 2012), Vol. 25.

[49] L. Yang and A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, Neurocomputing **415**, 295 (2020).

[50] Surrogate optimization for global minimization of time-consuming objective functions - MATLAB, https://www.mathworks.com/help/gads/surrogateopt.html.

[51] Surrogate optimization algorithm - MATLAB, https://www.mathworks.com/help/gads/surrogate-optimization-algorithm.html.

[52] M. C. Mackey and L. Glass, Oscillation and chaos in physiological control systems, Science **197**, 287 (1977).

[53] H. Wernecke, B. Sándor, and C. Gros, Chaos in time delay systems: An educational review, Phys. Rep. **824**, 1 (2019).

[54] Y. Kuramoto and T. Tsuzuki, Persistent propagation of concentration waves in dissipative media far from thermal equilibrium, Prog. Theor. Phys. **55**, 356 (1976).

[55] G. I. Sivashinsky, Nonlinear analysis of hydrodynamic instability in laminar ames I. derivation of basic equations, Acta Astron. **4**, 1177 (1977).

[56] R. A. Edson, J. E. Bunder, T. W. Mattner, and A. J. Roberts, Lyapunov exponents of the Kuramoto–Sivashinsky PDE, ANZIAM J. **61**, 270 (2019).

[57] Z. Liu and Y.-C. Lai, Coherence Resonance in Coupled Chaotic Oscillators, Phys. Rev. Lett. **86**, 4737 (2001).

[58] Y.-C. Lai and Z. Liu, Noise-enhanced temporal regularity in coupled chaotic oscillators, Phys. Rev. E **64**, 066202 (2001).

[59] A. S. Pikovsky and J. Kurths, Coherence Resonance in a Noise-Driven Excitable System, Phys. Rev. Lett. **78**, 775 (1997).

[60] J. D. Pelletier, Coherence resonance and ice ages, J. Geophys. Res. Atmos. **108**, 4645 (2003).

[61] J. L. A. Dubbeldam, B. Krauskopf, and D. Lenstra, Excitability and coherence resonance in lasers with saturable absorber, Phys. Rev. E **60**, 6580 (1999).

[62] C. Y. Lee, W. Choi, J.-H. Han, and M. S. Strano, Coherence resonance in a single-walled carbon nanotube ion channel, Science **329**, 1320 (2010).

[63] R. Benzi, A. Sutera, and A. Vulpiani, The mechanism of stochastic resonance, J. Phys. A **14**, L453 (1981).

[64] R. Benzi, G. Parisi, A. Sutera, and A. Vulpiani, A theory of stochastic resonance in climatic-change, J. Appl. Math. **43**, 565 (1983).

[65] B. McNamara and K. Wiesenfeld, Theory of stochastic resonance, Phys. Rev. A **39**, 4854 (1989).

[66] F. Moss, D. Pierson, and D. O'Gorman, Stochastic resonance: tutorial and update, Int. J. Bifurcation Chaos **04**, 1383 (1994).

[67] P. C. Gailey, A. Neiman, J. J. Collins, and F. Moss, Stochastic Resonance in Ensembles of Nondynamical Elements: The Role of Internal Noise, Phys. Rev. Lett. **79**, 4701 (1997).

[68] L. Gammaitoni, P. Hänggi, P. Jung, and F. Marchesoni, Stochastic resonance, Rev. Mod. Phys. **70**, 223 (1998).

[69] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, SmoothGrad: removing noise by adding noise, arXiv:1706.03825.

[70] B. Xu, C. Yang, and Z. Shi, Reinforcement learning output feedback NN control using deterministic learning technique, IEEE Trans. Neural Netw. Learning Syst. **24**, 635 (2013).

[71] R. Kamalapurkar, P. Walters, and W. E. Dixon, Model-based reinforcement learning for approximate optimal regulation, Automatica **64**, 94 (2016).

[72] M. Sangiorgio, F. Dercole, and G. Guariso, Forecasting of noisy chaotic systems with deep neural networks, Chaos Solit. Fractals **153**, 111570 (2021).

[73] Y. Grandvalet, S. Canu, and S. Boucheron, Noise injection: Theoretical prospects, Neural Comput. **9**, 1093 (1997).

[74] F. Wyffels, B. Schrauwen, and D. Stroobandt, Stable output feedback in reservoir computing using ridge regression, in *International Conference on Artificial Neural Networks* (Springer, Berlin, 2008), pp. 808–817.

[75] K.-C. Jim, C. L. Giles, and B. G. Horne, An analysis of noise in recurrent neural networks: convergence and generalization, IEEE Trans. Neural Netw. **7**, 1424 (1996).