Long-term prediction of chaotic systems with machine learning

Huawei Fan,[1,2] Junjie Jiang [●],[1] Chun Zhang [●],[1,2] Xingang Wang [●],[2] and Ying-Cheng Lai [●][1,3,*]

[1]*School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, Arizona 85287, USA*
[2]*School of Physics and Information Technology, Shaanxi Normal University, Xi'an 710062, China*
[3]*Department of Physics, Arizona State University, Tempe, Arizona 85287, USA*

Reservoir computing systems, a class of recurrent neural networks, have recently been exploited for model-free, data-based prediction of the state evolution of a variety of chaotic dynamical systems. The prediction horizon demonstrated has been about half dozen Lyapunov time. Is it possible to significantly extend the prediction time beyond what has been achieved so far? We articulate a scheme incorporating time-dependent but sparse data inputs into reservoir computing and demonstrate that such rare "updates" of the actual state practically enable an arbitrarily long prediction horizon for a variety of chaotic systems. A physical understanding based on the theory of temporal synchronization is developed.

A recently emerged interdisciplinary field is a machine-learning-based, model-free prediction of the state evolution of nonlinear/chaotic dynamical systems [1–12]. A paradigm that has been exploited is reservoir computing [13–16], a class of recurrent neural networks. Starting from the same initial condition, a well-trained reservoir system can generate a trajectory that stays close to that of the target system for a finite amount of time, realizing a short-term prediction. Because of the hallmark of chaos-sensitive dependence on initial conditions, the solution of the reservoir system will diverge from that of the original system exponentially. Nonetheless, if training is done properly so that the single-step error is orders of magnitude smaller than the oscillation range of the chaotic signal [12], an accurate prediction can be achieved in a short time. So far, the prediction horizon achieved is about five or six Lyapunov time [7], where one Lyapunov time is the inverse of the maximum Lyapunov exponent.

Is it possible to extend significantly the prediction horizon of reservoir computing? We provide an affirmative answer in this Rapid Communication. The key observation is that, after training, prediction is enabled because the neural network system can replicate the dynamical evolution of the target system (or synchronize with it) but only for a transient period of time. In the conventional scheme, data from the target system are used only during the training phase. A solution to extend the transient time is to provide some "update" of the target system. We thus conceive the scenario where, after the initial training, infrequent or sparse updates in the form of new measurements or the observation of the target system are available. We demonstrate that even rare updates of the

actual state enable an arbitrarily long prediction horizon to be achieved for a variety of chaotic systems. Essentially, before the trajectories of the reservoir and original systems diverge substantially (e.g., about to exceed a predefined accuracy), we correct the state of the reservoir system with a real measurement of a duration as small as a single data point. We develop a physical understanding based on the theory of temporal synchronization. Practically, with rare data updates, the reservoir computing system can replicate the evolution of the original system within some desired accuracy for an arbitrarily long time, in spite of chaos. This will have applications in fields where chaos arises.

The basic working of reservoir computing can be described briefly as follows. Suppose time series data represented by a relatively low-dimensional data vector from the target system to be predicted are available. As shown in Fig. 1, one feeds the time series into the input-to-reservoir (I/R) module to generate a time-dependent data vector whose dimension is significantly larger than that of the original data vector. The high-dimensional vector is then sent to a complex network constituting the core of the reservoir, whose size matches the dimension of the vector. That is, there is a one-to-one correspondence between any component of the high-dimensional vector and a node in the network, and the data from a component are fed into the corresponding node. The state of the reservoir network is updated according to some nonlinear function, and the resulting state vector is sent to a reservoir-to-output (R/O) module whose function is opposite that of the I/R module, i.e., to convert the high-dimensional vector of the reservoir back into a vector of the same low dimension as that of the original data from the target system. The reservoir network, once chosen, is fixed, so all parameters associated with it are hyperparameters. Training is done through a relatively small set of adjustable parameters associated with the R/O module, which can be tuned (or "learned") based on the available input data. During the training phase, the system is open as it requires input from the target system. After training, one feeds the output from R/O directly into
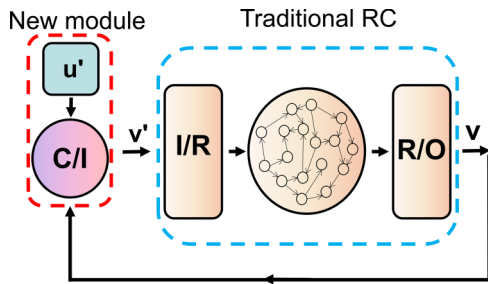
---

*Ying-Cheng.Lai@asu.edu

FIG. 1. Proposed reservoir computing (RC) system with rare state updates in the prediction phase, which is capable of generating an arbitrarily long prediction of the state evolution of any chaotic system. The I/R, reservoir, and R/O modules within the blue dashed box constitute the conventional reservoir computing system. Initial training with data from the target chaotic system is done in a conventional manner. The articulated scheme of rare state updates is represented by the C/I module inside the red dashed box, which couples sparse measurement data with the output of the system to generate updated inputs to the system.

the I/R module, closing the system. The system then evolves by itself. Some key numbers involved in reservoir computing for model-free predictions are as follows. For example, for the Kuramoto-Sivashinsky equation (KSE) with spatiotemporally chaotic solutions [7], the typical number of the spatial measurement sites (the dimension of the input vector) is 64 and the size of the reservoir network is about 5000. The number of parameters in the R/O module to be trained is 4992×64.

Our proposed reservoir computing system functioning in the prediction phase leading to an arbitrarily long prediction horizon is shown schematically in Fig. 1, where the three modules inside the blue dashed box represent the conventional system and the one inside the red dashed box is a new module incorporating rare state updates. The I/R module is described by W_{in}, a $D_r \times D_{in}$ random matrix that maps a D_{in}-dimensional input vector v into a D_r-dimensional vector $r(t)$, where $D_{in} \ll D_r$. The elements of W_{in} are generated from a uniform distribution in $[-\sigma, \sigma]$. The reservoir is a complex network of D_r nodes with an average degree d, whose connecting topology is described by the $D_r \times D_r$ matrix A. (For simplicity, we choose it to be a directed random network.) A recent work [12] has revealed that successful training and finite-time prediction can be achieved if the spectral radius of the network is in a finite range, which can be adjusted by properly normalizing the link weights in the network. The R/O module is represented by a $D_{out} \times D_r$ matrix, whose elements are parameters determined through training [3,7]. Typically, we have $D_{out} = D_{in}$. Without any state update, in the prediction phase, the reservoir computing system is a self-evolving dynamical system described by $r(t + \Delta t) = \tanh[A \cdot r(t) + W_{in} \cdot v(t)]$ and $v(t + \Delta t) = W_{out} \cdot f[r(t + \Delta t)]$, where $f(r)$ is the output function [3,7]: $f_i(r) = r_i$ and $f_i(r) = r_i^2$ for odd and even index i ($i = 1, 2, \ldots, D_r$), respectively. Associated with the dynamical evolution of the reservoir system are two sets of dynamical variables: the high-dimensional reservoir state vector r and the (typically) low-dimensional output vector v. The matrices W_{in} and A are predefined while W_{out} is determined by training during which v is replaced by the state vector u of the target system. After training is completed, to set the

initial condition for the reservoir network, one approach is to continue to use u in place of v but only for a few time steps, after which the reservoir system executes natural dynamical evolution by itself.

In the conventional scheme [1–12], after withdrawing the true state vector u so that the system is closed, real measurements are no longer used, resulting in a relatively short prediction horizon for chaotic systems because of the exponential divergence between the trajectories of the reservoir and true systems. Our idea, as shown in Fig. 1, is to update the reservoir state with a sparsely sampled real state vector u' before the divergence exceeds a predefined tolerance limit, i.e., the updates are needed only rarely. In particular, during the update, the input to the I/R module can be written as $v'(t) = v(t) + c[u'(t) - v(t)]$, where u' contains data at t and c is the coupling parameter. Most of the time during the evolution, we still have $v'(t) = v(t)$. Updating is effectively an on-off coupling process between the reservoir and the true systems, where the "on" phase is significantly more sparse than the "off" phase. Prediction can then be viewed as a synchronization process [11,17] between the two systems that are coupled but only intermittently [18,19].

We test the predictive power of our proposed reservoir computing scheme with a large number of chaotic systems. Here, we present two examples of high-dimensional chaotic systems: KSE and the complex Ginzburg-Landau equation (cGLE). (Examples of a number of low-dimensional chaotic systems are presented in the Supplemental Material [20].) The KSE is $y_t + yy_x + y_{xx} + y_{xxxx} = 0$, where $y(x, t)$ is a scalar field in the interval $x \in (0, L)$ with periodic boundaries. We divide the spatial domain into M uniform subintervals. Figure 2(a) shows a typical spatiotemporal chaotic solution for $L = 22$, where the numerical integration parameters are $M = 64$ and $\Delta t = 0.25$, and the maximum Lyapunov exponent is $\Lambda_{max} \approx 0.05$. Thus, approximately 80 time steps correspond to one Lyapunov time. The M-dimensional data vector is fed into the reservoir computing system with parameters $D_{in} = D_{out} = M$, $D_r = 4992$, $\sigma = 1$, $d = 3$, and $\rho = 0.1$. In addition, to avoid overfitting of W_{out} during the training process, we set the relevant bias parameter [7] to be $\eta = 1 \times 10^{-4}$. Figure 2(b) shows the difference between the evolution of the reservoir and true system, i.e., the prediction error (color coded). The prediction horizon is about five Lyapunov time.

We now instigate rare updates with the true data $y_{act}(x, t)$, where the coupling function is $y'_{rc}(x, t) = y_{rc}(x, t) + c[y_{act}(x, t) - y_{rc}(x, t)]$ if there is a data update at the space-time point (x, t), otherwise we have $y'_{rc}(x, t) = y_{rc}(x, t)$, where y_{rc} is the output of the reservoir system. We divide the output time series into equal time intervals, each of T time steps. In one interval, the actual data points are available for consecutive T_0 steps, and there is no update for the remaining $T - T_0$ steps. In the space, we select M_c uniform spatial points from the total M measurement points. For illustration, we $T_0 = 1$ and $M_c = 64$, i.e., each component of the input vector to the reservoir system (corresponding to a distinct measurement point in space) receives one true data point every T time steps. Figures 2(c)–2(e) show the prediction error for $T = 240, 160$, and 80 time steps, respectively, corresponding to three, two, and one Lyapunov time. For $T = 240$ [Fig. 2(c)],
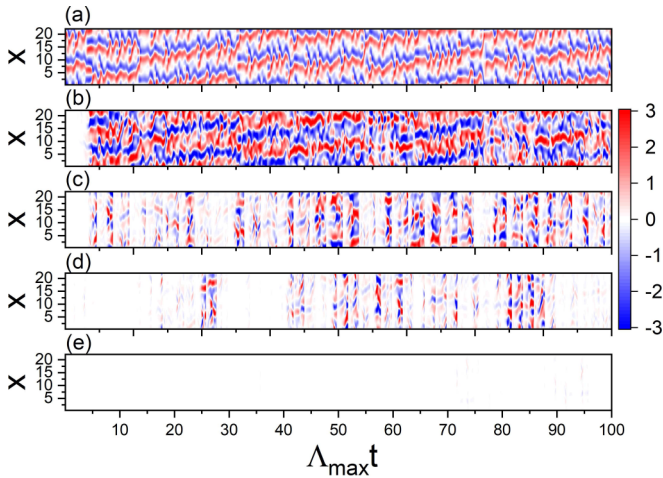
FIG. 2. Prediction of spatiotemporal chaotic solution of KSE. The system size is $L = 22$, the number of measurement points is $M = 64$, and the value of the coupling parameter is $c = 1.0$. (a) True spatiotemporal evolution of the chaotic solution. (b) The difference (error) between the predicted and true solutions without any measurement updating, i.e., with updating period $T = \infty$. The prediction horizon is about five Lyapunov time. (c)–(e) The error with updating period $T = 240, 160$, and 80 time steps, corresponding to three, two, and one Lyapunov time, respectively, where the update consists of a single data point from the true system. In (e), error at all times is below the tolerance, signifying an arbitrarily long prediction horizon.

the error is reduced as compared the case of no data update [Fig. 2(b)] and exhibits intermittency, implying an intermittently synchronous behavior between the reservoir and the true system. In this case, the time interval in which the error is below the tolerance emerges intermittently in time. As the updating becomes more frequent, the small-error intervals are generally enlarged, as shown in Fig. 2(d) for $T = 160$. Remarkably, for $T = 80$, the error is essentially zero in the whole time interval considered. This means that, insofar as a single true data point is used to update the reservoir system in every Lyapunov time, the prediction horizon becomes arbitrarily long.

To obtain a systematic picture of the prediction horizon, we calculate the average error δe over a long time interval, e.g., about 100 Lyapunov time, in the parameter plane (T_0, T) (for $T_0 \leqslant T$). Figure 3(a) shows, for $M_c = M/2 = 32$ (the number of spatial coupling channels) and $c = 2.0$, the error behavior. In most of the parameter region, the error is small: $\delta e < 0.01$. Long-term prediction fails only for extremely rare updates, i.e., for small values of T_0 and large values of T. Fixing the value of T_0, we obtain the curves of δe vs T. For $T_0 = 1$ (black line), δe increases rapidly with T. As more data points are included in each update, e.g., as the value of T_0 is increased from two to ten, δe decreases gradually. Figure 3(c) shows that, for $M_c = M = 64$, the error remains small in all cases, giving rise to a significantly augmented prediction horizon. Figure 3(d) shows the prediction error for T up to 500 time steps. For a change from $T_0 = 1$ to $T_0 = 2$, error δe decreases sharply. However, for $T_0 \geqslant 2$, the prediction
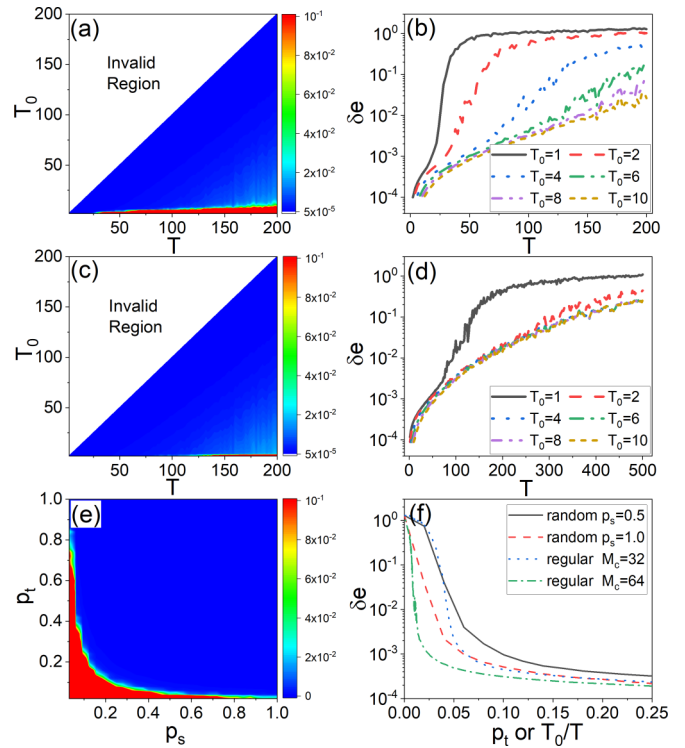

FIG. 3. Behaviors of prediction error for KSE. (a) Prediction error in the parameter plane (T_0, T) for $M_c = 32$ and $c = 2.0$. (b) For $M_c = 32$, prediction error δe vs T for different values of T_0. (c) Prediction error for $M_c = 64$ and $c = 1.0$. (d) For $M_c = 64$, δe vs T for different values of T_0. (e) For random updating, prediction error in the parameter plane (p_t, p_s) for $c = 1.2$ and (f) δe vs p_t (or T_0/T).

error becomes saturated. Comparing Fig. 3(b) with 3(d), we observe a dramatic reduction in δe for $M_c = 64$.

The cases considered so far are for rare but regular updates. What about random updates? We examine the parameter plane (p_t, p_s), where p_t is the probability of update in each time step and p_s is the probability for a spatial point to receive updates. Figure 3(e) shows the average prediction error, which exhibits an approximately symmetric behavior with respect to p_t and p_s. Figure 3(f) shows, for fixed $p_s = 0.5$ and $p_s = 1$ (corresponding to the cases of $M_c = 32$ and $M_c = 64$ with regular coupling, respectively), random updating yields somewhat larger errors than those with regular coupling.

We now demonstrate the predictive power of our reservoir computing scheme for the 1D cGLE in the regime of spatiotemporal chaos, $A_t = (1 + i\alpha)A_{xx} + A - (1 + i\beta)|A|^2 A$, where $A(x, t)$ is a complex field in the interval $x \in (-L/2, L/2)$ with the periodic boundary condition, and α and β are parameters. The cGLE is a general model for a variety of physical phenomena [21–23]. For $L = 18$, $\alpha = 2$, and $\beta = -2$, the 1D cGLE exhibits spatiotemporal chaos with the maximum Lyapunov exponent $\Lambda \approx 0.23$. We divide the whole interval into $M = 32$ equally spaced points—the measurement sites. For integration step $\Delta t = 0.07$, one Lyapunov time corresponds to about 65 time steps. Figure 4(a) shows the true spatiotemporal evolution pattern $|A(x, t)|$. Because the field $A(x, t)$ is complex, it is necessary to use both the real
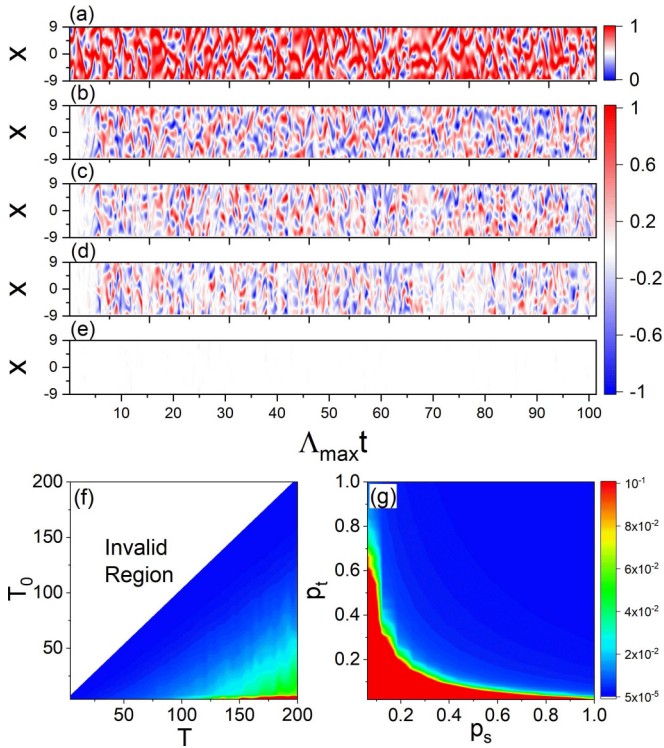
FIG. 4. Long time prediction of the state evolution of one-dimensional cGLE. The system size is $L = 18$ and the number of measurement sites is $M = 32$. (a) Spatiotemporal pattern of true state evolution. (b)–(e) The difference between the predicted and actual state evolution for $T = \infty$, $T = 195$, $T = 130$, and $T = 65$, respectively. (f) For regular updating, the prediction error in the parameter plane (T_0, T) for $M_c = 32$ and $c = 0.4$. (g) For random updating, the prediction error in the parameter plane (p_t, p_s) for $c = 0.4$.

$[A^r(x, t)]$ and imaginary $[A^i(x, t)]$ parts for training the reservoir system. The parameters of the reservoir system are $D_{\text{in}} = D_{\text{out}} = 2M$, $D_r = 9984$, $\sigma = 1$, $d = 3$, $\rho = 0.1$, and $\eta = 2 \times 10^{-5}$. Without any update, the error between the predicted and true state evolution is shown in Fig. 4(b), where the prediction horizon is about five Lyapunov time. To introduce rare updates of the true state, we use the coupling scheme $A_{rc}^{\prime r(i)}(x, t) = A_{rc}^{r(i)}(x, t) + 0.4[A_{act}^{r(i)}(x, t) - A_{rc}^{r(i)}(x, t)]$ if there is actual data at the space-time point (x, t), and otherwise $A_{rc}^{\prime r(i)}(x, t) = A_{rc}^{r(i)}(x, t)$, where $A_{act}^{r(i)}(x, t)$ is the real (imaginary) part of the true data and $A_{rc}^{r(i)}$ is the real (imaginary) part of the predicted state. We demonstrate updating with only a single data point, $T_0 = 1$. Figures 4(c)–4(e) show the prediction error for $T = 195$, 130, and 65, corresponding to three, two, and one Lyapunov time, respectively. Evidence of intermittent synchronization between the reservoir and the actual systems is shown in Figs. 4(c) and 4(d). As for the spatiotemporally chaotic KSE system, updating the reservoir system with a single data point every Lyapunov time [Fig. 4(e)] leads to an arbitrarily long prediction horizon for the spatiotemporally chaotic state of the 1D cGLE.

Figures 4(f) and 4(g) show the time-averaged prediction error δe in the parameter planes (T_0, T) (regular updating scheme) and (p_s, p_t) (random updating scheme), respectively,

where the average is taken over approximately 100 Lyapunov time. For regular updating, a small prediction error can be achieved in most of the parameter plane. For random updating, the error decreases with an increase in p_t and/or p_s.

The theoretical explanation for the observed long-term prediction is that, after proper training the output vector of the reservoir system follows the state vector of the target system for a finite amount of time, indicating complete synchronization between the two systems. Without state updating, the synchronization state is slightly unstable, leading to a short prediction horizon. State updates, even applied rarely, represent a kind of perturbation that makes the synchronization state less unstable, prolonging the prediction horizon. When the frequency of the updates is such that there is one update within one Lyapunov time, the synchronization state becomes stable, giving rise to an arbitrarily long prediction horizon. This scenario has been verified using low-dimensional chaotic systems [20] through a synchronization stability analysis [24,25]. We also note that, for a variety of low-dimensional chaotic systems including the classic Rössler [26] and Lorenz [27] oscillators, the Hindmarsh-Rose neuron [28], and a chaotic food web [29], rare updates to a single state variable enables a properly trained reservoir system to predict *all* state variables for an arbitrarily long period of time [20]. For example, for the chaotic food web, it is necessary only to supply sparse vegetation data for the reservoir system to correctly predict the abundances of the herbivores and predators, for as long as one wishes.

To summarize, existing reservoir computing systems can predict the dynamical evolution of chaotic systems but only for a short period of time. We have articulated a scheme incorporating true state updates and demonstrated that rare updates of a subset of state variables can significantly prolong the prediction horizon. Of particular interest is the finding that, insofar as there is a state update of a single data point within one Lyapunov time, the prediction horizon can be made arbitrarily long. The machine-learning scheme proposed and studied here has the potential to extend significantly the application scope of reservoir computing in predicting complex dynamical systems.

The difference in our work is threefold: (1) We have introduced intermittent data updates into machine learning, (2) we have demonstrated that a long prediction time can be achieved, and (3) we have introduced the concept of temporal synchronization with on-off coupling to understand the working of the reservoir computing machine with state updating. Our work sheds light on the working of reservoir computing in predicting the state evolution of chaotic systems. In particular, in the synchronization-based scenario, the predictability of reservoir computing is the result of its ability to synchronize with the target chaotic system for a finite amount of time. At the start of the prediction phase, the reservoir system has the same initial condition as the target system. Because of temporal synchronizability, the reservoir system is able to follow the target system for some time before the synchronization error becomes significant. Without state updates, the time it takes for the error to grow to a predefined threshold value determines the prediction time. State updates, even being rare, reset the synchronization error from time to time, insofar as a new update is provided before the error exceeds the threshold.

In essence, the basic idea of our method is similar to that of data assimilation, where models and measurements (true state updates) are combined to generate accurate predictions with applications in, e.g., weather forecasting [30–32]. In a recent study, data assimilation and machine learning have been combined to emulate the Lorenz 96 model from sparse and noisy observations [33]. It is also noteworthy that, in predicting chaotic dynamical systems, an alternative machine-learning-based framework is radial basis function networks—artificial neural networks employing radial basis functions as activation functions [34,35]. Given a set of inputs, such a network outputs a signal that is a linear combination of radial basis functions of the inputs, and the parameters of the artificial neurons are determined through training based on, e.g., the standard backpropagation scheme. The method has been demonstrated to be effective for low-dimensional chaotic systems such as the logistic map and the classic Lorenz chaotic oscillator [36–39]. Whether the method can be superior to reservoir computing in predicting spatiotemporal chaotic systems is an open question.

H.F. and J.J. contributed equally to this work.

[1] 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).

[2] 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).

[3] 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).

[4] Z. 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).

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

[6] J. Pathak, A. Wilner, R. Fussell, S. Chandra, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model, Chaos **28**, 041101 (2018).

[7] 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).

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

[9] K. Nakai and Y. Saiki, Machine-learning inference of fluid variables from data using reservoir computing, Phys. Rev. E **98**, 023111 (2018).

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

[11] T. Weng, H. Yang, C. Gu, J. Zhang, and M. Small, Synchronization of chaotic systems and their machine-learning models, Phys. Rev. E **99**, 042203 (2019).

[12] 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).

[13] H. Jaeger, The "echo state" approach to analyzing and training recurrent neural networks—with an erratum note, Bonn, Germany: German National Research Center for Information Technology GMD Technical Report No. 148, p. 13, 2001 (unpublished).

[14] W. Mass, 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).

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

[16] 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).

[17] Z. Lu, B. Hunt, and E. Ott, Attractor reconstruction by machine learning, Chaos **28**, 061104 (2018).

[18] L. Chen, C. Qiu, and H. B. Huang, Synchronization with on-off coupling: Role of time scales in network dynamics, Phys. Rev. E **79**, 045101(R) (2009).

[19] S. Li, N. Sun, L. Chen, and X. Wang, Network synchronization with periodic coupling, Phys. Rev. E **98**, 012304 (2018).

[20] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevResearch.2.012080 for three examples of the long-term prediction of low-dimensional chaotic systems and a heuristic explanation of the ability for reservoir computing to achieve an arbitrarily long prediction horizon based on the synchronization mechanism .

[21] I. S. Aranson and L. Kramer, The world of the complex Ginzburg-Landau equation, Rev. Mod. Phys. **74**, 99 (2002).

[22] M. C. Cross and P. C. Hohenberg, Pattern formation outside of equilibrium, Rev. Mod. Phys. **65**, 851 (1993).

[23] Y. Kuramoto, *Chemical Oscillations, Waves and Turbulence* (Springer, Berlin, 1984).

[24] J. F. Heagy, L. M. Pecora, and T. L. Carroll, Short Wavelength Bifurcations and Size Instabilities in Coupled Oscillator Systems, Phys. Rev. Lett. **74**, 4185 (1995).

[25] L. M. Pecora and T. L. Carroll, Master Stability Functions for Synchronized Coupled Systems, Phys. Rev. Lett. **80**, 2109 (1998).

[26] O. E. Rössler, Equation for continuous chaos, Phys. Lett. A **57**, 397 (1976).

[27] E. N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. **20**, 130 (1963).

[28] J. L. Hindmarsh and R. M. Rose, A model of neuronal bursting using three coupled first order differential equations, Proc. R. Soc. London, Ser. B **221**, 87 (1984).

[29] B. Blasius, A. Huppert, and L. Stone, Complex dynamics and phase synchronization in spatially extended ecological systems, Nature (London) **399**, 354 (1999).

[30] D. J. Patil, B. R. Hunt, E. Kalnay, J. A. Yorke, and E. Ott, Local Low Dimensionality of Atmospheric Dynamics, Phys. Rev. Lett. **86**, 5878 (2001).

[31] E. Ott, B. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. J. Patil, and J. A. Yorke, A local ensemble Kalman filter for atmospheric data assimilation, Tellus A **56**, 415 (2004).

[32] I. Szunyogh, E. Kostelich, G. Gyarmati, D. J. Patil, B. R. Hunt, E. Kalnay, E. Ott, and J. A. Yorke, Assessing a local ensemble Kalman filter: Perfect model experiments with the national centers for environmental prediction global model, Tellus A **57**, 528 (2005).

[33] J. Brajard, A. Carrassi, M. Bocquet, and L. Bertino, Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the Lorenz 96 model, Geosci. Model Dev. Discuss. (2019), doi:10.5194/gmd-2019-136.

[34] D. S. Broomhead and D. Lowe, Multivariable functional interpolation and adaptive networks, Complex Syst. **2**, 321 (1988).

[35] T. Poggio and F. Girosi, Networks for approximation and learning, Proc. IEEE **78**, 1484 (1990).

[36] A. Gholipour, B. N. Araabi, and C. Lucas, Predicting chaotic time series using neural and neurofuzzy models: A comparative study, Neural Process. Lett. **24**, 217 (2006).

[37] D. Chen and W. Han, Prediction of multivariate chaotic time series via radial basis function neural network, Complexity **18**, 55 (2013).

[38] M. K. Rafsanjani and M. Samareh, Chaotic time series prediction by artificial neural networks, J. Comput. Methods Sci. Eng. **16**, 599 (2016).

[39] V. T. Nguyen and T. A. Duong, Chaotic time series prediction using radial basis function networks, in *Proceedings of the 2018 4th International Conference on Green Technology and Sustainable Development (GTSD)* (IEEE, Piscataway, NJ, 2018).

Supplementary Information for

Long-term prediction of chaotic systems with machine learning

Huawei Fan, Junjie Jiang, Chun Zhang, Xingang Wang, and Ying-Cheng Lai*

Corresponding author email: Ying-Cheng.Lai@asu.edu

CONTENTS

1

I. MECHANISM FOR RESERVOIR COMPUTING TO ACHIEVE ARBITRARILY LONG PREDICTION HORIZON - A HEURISTIC ANALYSIS BASED ON SYNCHRONIZATION

To understand the mechanism behind the realization of long prediction horizon in our proposed reservoir computing scheme with state updating, we resort to synchronization analysis. To facilitate the analysis, we apply our scheme to the chaotic Rössler oscillator given by [1]:

$$
\begin{aligned}
\dot{x} &= -y - z, \\
\dot{y} &= x + 0.2y, \\
\dot{z} &= 0.2 + (x - 9.0)z.
\end{aligned}
\tag{S1.1}
$$

We obtain the three time series $[x(t), y(t), z(t)]$ at the step size $\Delta t = 0.05$. The values of the parameters of the reservoir computing system are $D_{in} = D_{out} = 3$, $D_r = 600$, $\sigma = 0.15$, $d' = 0.2$, $\rho = 0.2$, and $\eta = 1 \times 10^{-7}$, where d' is the link density of the complex neural network. The elements of the $D_r \times D_{in}$ input matrix $\boldsymbol{W}_{in}$ are generated from the uniform distribution $[-\sigma, \sigma]$. The non-zero elements of the $D_r \times D_r$ matrix $\boldsymbol{A}$ are generated from the uniform distribution $[-1, 1]$. Figure S1(a) shows the prediction result with the conventional scheme, where the prediction horizon is about $t \approx 75$ (corresponding to approximately 15 average cycles of oscillation). Figure S1(b) shows that, with our proposed scheme, the prediction horizon is practically infinite, where an update of the actual data of *a single dynamical variable*, y_{act}, is coupled into the system ($T_0 = 1$ and $T = 50$) once every 50 time steps. More specifically, in the iterative process of reservoir computing system, the input data y_{rc} is replaced by $y'_{rc} = y_{rc} + c(y_{act} - y_{rc})$ once every 50 time steps for $c = 0.8$. Note that, in Fig. S1(b), only the true and predicted x time series are shown, but time series from the other two dynamical variables give essentially the same prediction result. Figure S1(c) shows the color-coded average prediction error δe in the parameter plane (T_0, T), where there are multiple parameter regions in which the error is small. The patterns in Fig. S1(c) are reminiscent of the phenomenon of ragged synchronization in coupled chaotic oscillator systems [2, 3], suggesting the use of synchronization theory to understand the working mechanism of our articulated reservoir computing machine.

A well trained reservoir computing system can be viewed as a high-dimensional replica of the target system. A previous calculation of the Lyapunov exponents of the reservoir dynamical network revealed that the first few exponents are indeed approximate values of the exponents of the true system, and the vast set of remaining exponents have large negative values [4]. This is anticipated as the large negative exponents are necessary to reduce the exceedingly high dimension of the reservoir network to the low-dimensional target system through a strong compression of the dynamics along vast majority of orthogonal directions in the phase space. For the chaotic Rössler system, the considerations suggest that the dynamics of the reservoir computing system be approximately described by

$$
\begin{aligned}
\dot{x}_{rc} &= -y_{rc} - z_{rc}, \\
\dot{y}_{rc} &= x_{rc} + 0.2y_{rc} + \varepsilon(t)(y_{act} - y_{rc}), \\
\dot{z}_{rc} &= 0.2 + (x_{rc} - 9)z_{rc},
\end{aligned}
\tag{S1.2}
$$

where, in the coupling term $\varepsilon(t)(y_{act} - y_{rc})$, $\varepsilon(t)$ specifies the on-off nature of the coupling: $\varepsilon(t) = \varepsilon$ if $nT\Delta t < t < T_0\Delta t + nT\Delta t$ ($n = 0, 1, 2, \ldots$) and $\varepsilon(t) = 0$ otherwise. There is a linear relation between ε and c: $c = \varepsilon\Delta t$. The data of y_{act} are generated from the target system
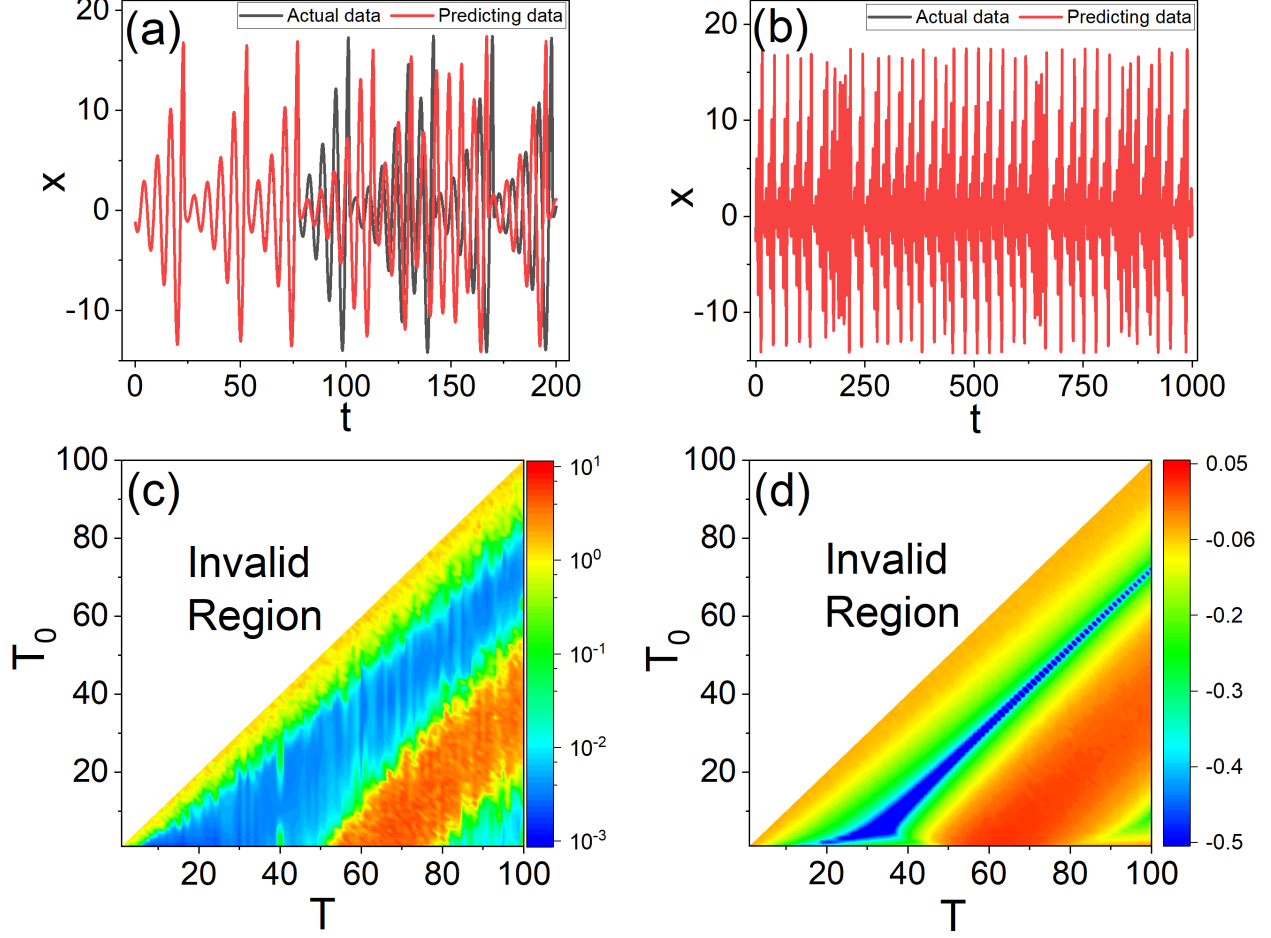
FIG. S1. *Predicting state evolution of chaotic Rössler oscillator and synchronization analysis.* (a) Result of prediction with conventional reservoir computing: true (black) and predicted (red) time series $x(t)$. The prediction horizon is approximately 15 average periods of oscillation as the true and predicted time series begin to diverge after this time. (b) True and predicted time series with our proposed reservoir computing scheme incorporating rare state updating. The two time series overlap completely and the prediction horizon is practically infinite. (c) Color-coded prediction error in the parameter plane (T_0, T), where T is the updating period (in units of Δt, the time step between two successive iterations of the reservoir neural network) and $T_0 < T$ is the number of time steps during which there is true data input. (d) The maximum transverse Lyapunov exponent in (T_0, T) from synchronization analysis. There is a qualitative correspondence between the small error [(c)] and negative Lyapunov exponent regions in the parameter plane.

Eq. (S1.1). In our scheme, predictability implies synchronization between the reduced reservoir computing system Eq. (S1.2) and the true system Eq. (S1.1) with only on-off coupling, where the "on" phase is typically significantly shorter than the "off" phase.

We use stability analysis to quantify synchronization. Let $\delta x = x_{rc} - x_{act}$, $\delta y = y_{rc} - y_{act}$, and $\delta z = z_{rc} - z_{act}$ be the infinitesimal perturbations transverse to the synchronization manifold. The variational equations can be obtained by linearizing Eq. (S1.2) with respect to the true state of the

target system as described by Eq. (S1.1):

$$\begin{aligned}
\delta\dot{x} &= -\delta y - \delta z, \\
\delta\dot{y} &= \delta x + 0.2\delta y - \varepsilon(t)\delta y, \\
\delta\dot{z} &= z\delta x + (x - 9.0)\delta z.
\end{aligned} \qquad (S1.3)$$

Combining Eqs. (S1.1) and (S1.3), we calculate the maximum transverse Lyapunov exponent Λ. Stable synchronization requires $\Lambda < 0$. Figure S1(d) shows the color-coded value of Λ in the parameter plane (T_0, T), which exhibits typical features of ragged synchronization [2, 3]. Comparing Fig. S1(c) with Fig. S1(d), we observe a striking degree of similarity, indicating synchronization between the reservoir computing and the target systems subject to on-off coupling as the dynamical mechanism responsible for realizing the long prediction horizon with rare data updating.
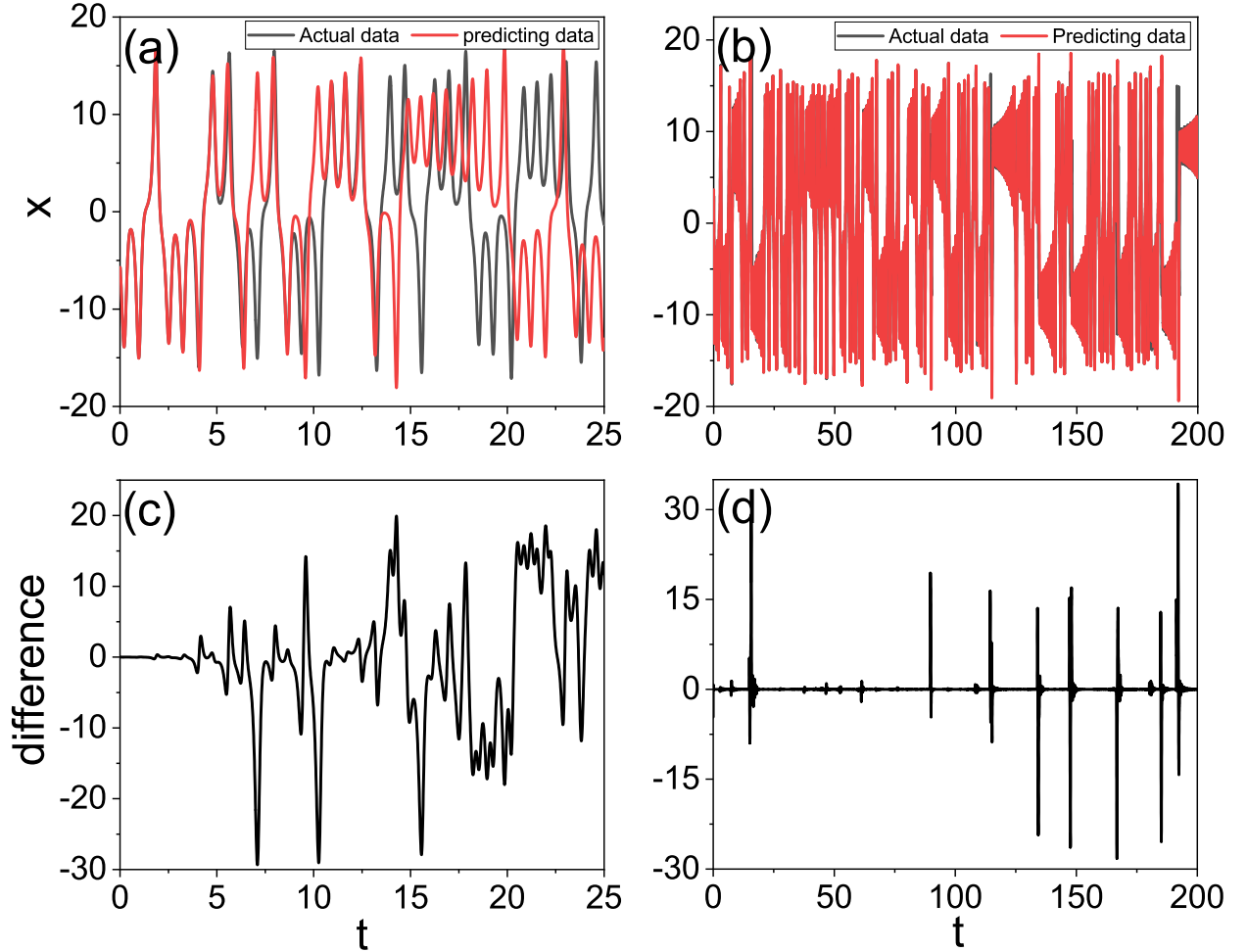

FIG. S2. *Predicting the chaotic Lorenz system with rare state updating.* (a) Prediction with the conventional reservoir computing scheme without state updating, where the prediction horizon is about ten oscillations. (b) Practically infinite prediction horizon achieved with sparse input of actual data for $T_0 = 1$, $T = 40$, and $c = 0.8$. (c,d) The difference between the prediction and actual time series corresponding to the results in (a,b), respectively.

4

II.   PREDICTING CHAOTIC LORENZ SYSTEM

We demonstrate that a practically infinite prediction horizon can be achieved for the classic chaotic Lorenz system [5] with rare state updating. The equations of the system are

$$
\begin{aligned}
\dot{x} &= 10(y - x), \\
\dot{y} &= x(28 - z) - y, \\
\dot{z} &= xy - 8/3z.
\end{aligned}
\tag{S2.4}
$$

We obtain the time series $x(t)$, $y(t)$, and $z(t)$ with integration step size $\Delta t = 0.01$. The parameter setting of the reservoir computing system is $D_{in} = D_{out} = 3$, $D_r = 600$, $\sigma = 0.1$, $d' = 0.3$, $\rho = 1.2$, and $\eta = 1 \times 10^{-5}$. Figure S2(a) shows the prediction result with the conventional scheme without any state updating, where the prediction horizon is $t \approx 5$ (corresponding to approximately ten average oscillations). Figure S2(b) shows that, with rare state updating ($T_0 = 1$ and $T = 40$) of one of the dynamical variables, mathematically represented as replacement of y_{rc} by $y'_{rc} = y_{rc} + c(y_{act} - y_{rc})$ once every 40 time steps, practically an arbitrarily long prediction horizon can be achieved. FiguresS2(c) and S2(d) show the difference between the predicted and true time series from the conventional and our proposed reservoir computing schemes, respectively. It can be seen that, with rare state updating, the prediction error is essentially zero for the time interval displayed, with relatively large errors occurring only at about a few dozen time steps (out of 2×10^4 time steps).

III.   PREDICTING CHAOTIC HINDMARSH-ROSE NEURON DYNAMICS

We test our reservoir computing scheme with rare state updating for the chaotic Hindmarsh-Rose neuron model [6]:

$$
\begin{aligned}
\dot{x} &= y + 3x^2 - x^3 - z + 3.2, \\
\dot{y} &= 1 - 5x^2 - y, \\
\dot{z} &= -0.006z + 0.024(x + 1.6),
\end{aligned}
\tag{S3.5}
$$

where x is the membrane potential, y and z are the transport rates of the fast and slow channels, respectively. The integration time step is $\Delta t = 0.1$. The three time series $x(t)$, $y(t)$, and $z(t)$ are used to train the reservoir computing system with parameters $D_{in} = D_{out} = 3$, $D_r = 600$, $\sigma = 0.6$, $d' = 0.2$, $\rho = 0.3$, and $\eta = 1 \times 10^{-7}$. Figure S3(a) shows the result with the conventional reservoir computing scheme without any state updating, where the prediction horizon is $t \approx 300$, in which there are about a dozen spiking events. With rare state updating (once every 100 time steps: $T_0 = 1$ and $T = 100$) of two state variables (x_{act} and y_{act}), the prediction horizon is practically infinite. Mathematically, the updating scheme can be described as a single replacement every 100 time steps of x_{rc} and y_{rc} by $x'_{rc} = x_{rc} + c(x_{act} - x_{rc})$ and $y'_{rc} = y_{rc} + c(y_{act} - y_{rc})$ respectively, for $c = 0.9$. The prediction errors corresponding to Figs. S3(a) and S3(b) are shown in Figs. S3(c) and S3(d), respectively.
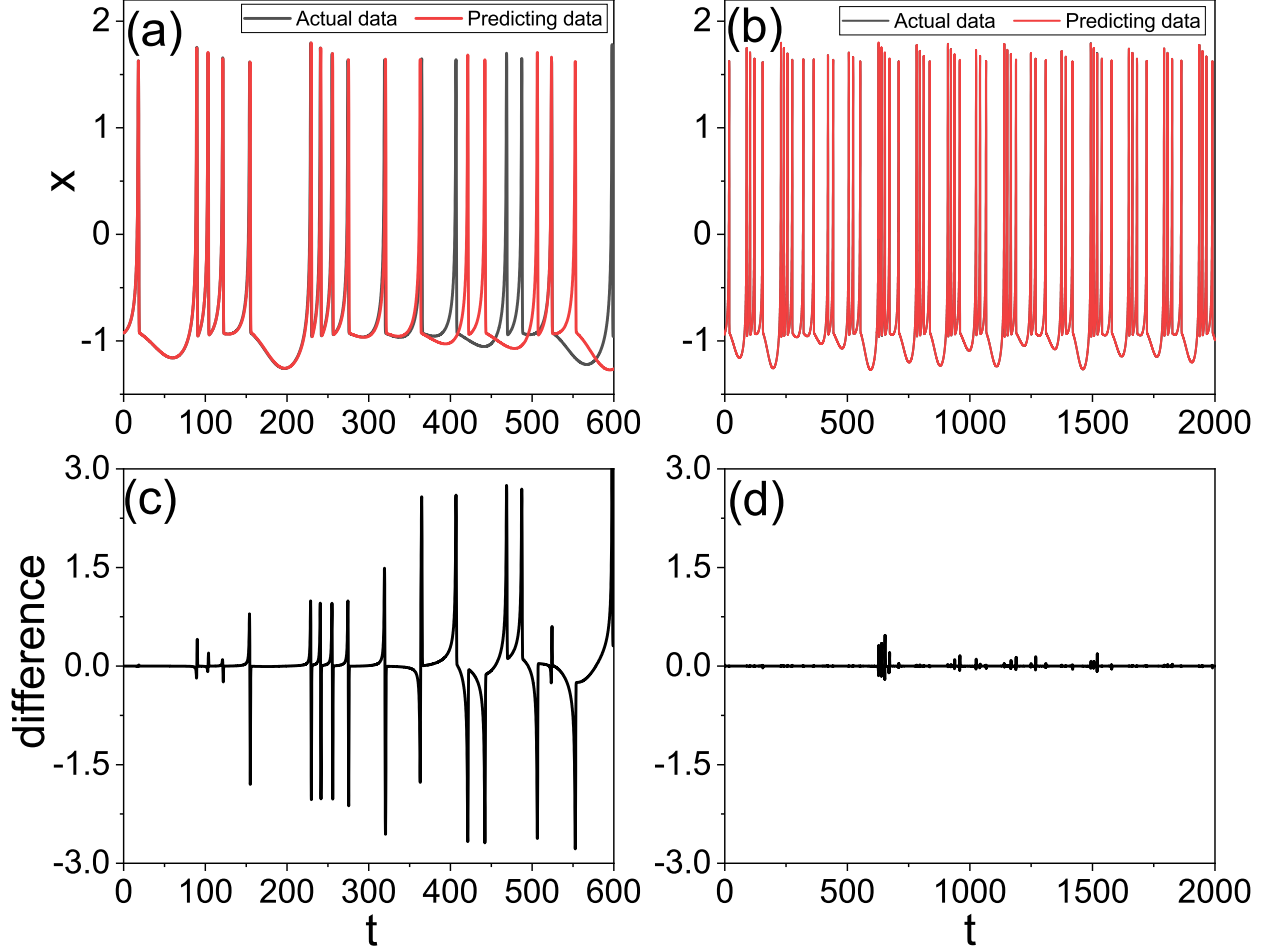
FIG. S3. *Predicting the state evolution of a chaotic Hindmarsh-Rose neuron.* (a) Prediction without state updating. (b) Arbitrarily long prediction horizon with rare state updating for $T_0 = 1$, $T = 300$, and $c = 0.9$. (c,d) The difference between the predicted and true state evolution for the cases in (a,b), respectively.

IV. PREDICTING POPULATION EVOLUTION IN A CHAOTIC FOOD WEB

We demonstrate an arbitrarily long prediction horizon for the following chaotic food web system [7]:

$$
\begin{aligned}
\dot{x} &= x - 0.2\frac{xy}{1 + 0.05x}, \\
\dot{y} &= -y + 0.2\frac{xy}{1 + 0.05x} - yz, \\
\dot{z} &= -10(z - 0.006) + yz,
\end{aligned}
\tag{S4.6}
$$

where x, y and z represent vegetation, herbivores and predators, and the evolution displays uniform phase evolution but with chaotic amplitude modulation. The integration step size is $\Delta t = 0.1$. The parameter values of the reservoir computing system are $D_{in} = D_{out} = 3$, $D_r = 600$, $\sigma = 0.15$, $d' = 0.2$, $\rho = 0.2$, and $\eta = 1 \times 10^{-7}$. Figure S4(a) shows the prediction result from the conventional scheme without any state updating, where the prediction horizon is $t \approx 50$ (containing seven or eight bursts in the predator population). In the food web system, vegetation data are

6

relatively easy to be collected, so we use x_{act} to perform rare state updating. Figure S4(b) shows the predicted and actual predator time series for $T_0 = 1$ and $T = 50$, i.e., we replace x_{rc} by $x'_{rc} = x_{rc} + c(x_{act} - x_{rc})$ once every 50 time steps. Visually the two types of time series cannot be distinguished. Figures S4(c) and S4(d) show the corresponding evolution of the prediction error for Figs. S4(a) and S4(b), respectively. We see that, with rare state updating, the prediction error is exceedingly small in the long time interval (10^4 time steps) tested, indicating that a practically infinite prediction horizon has been achieved.
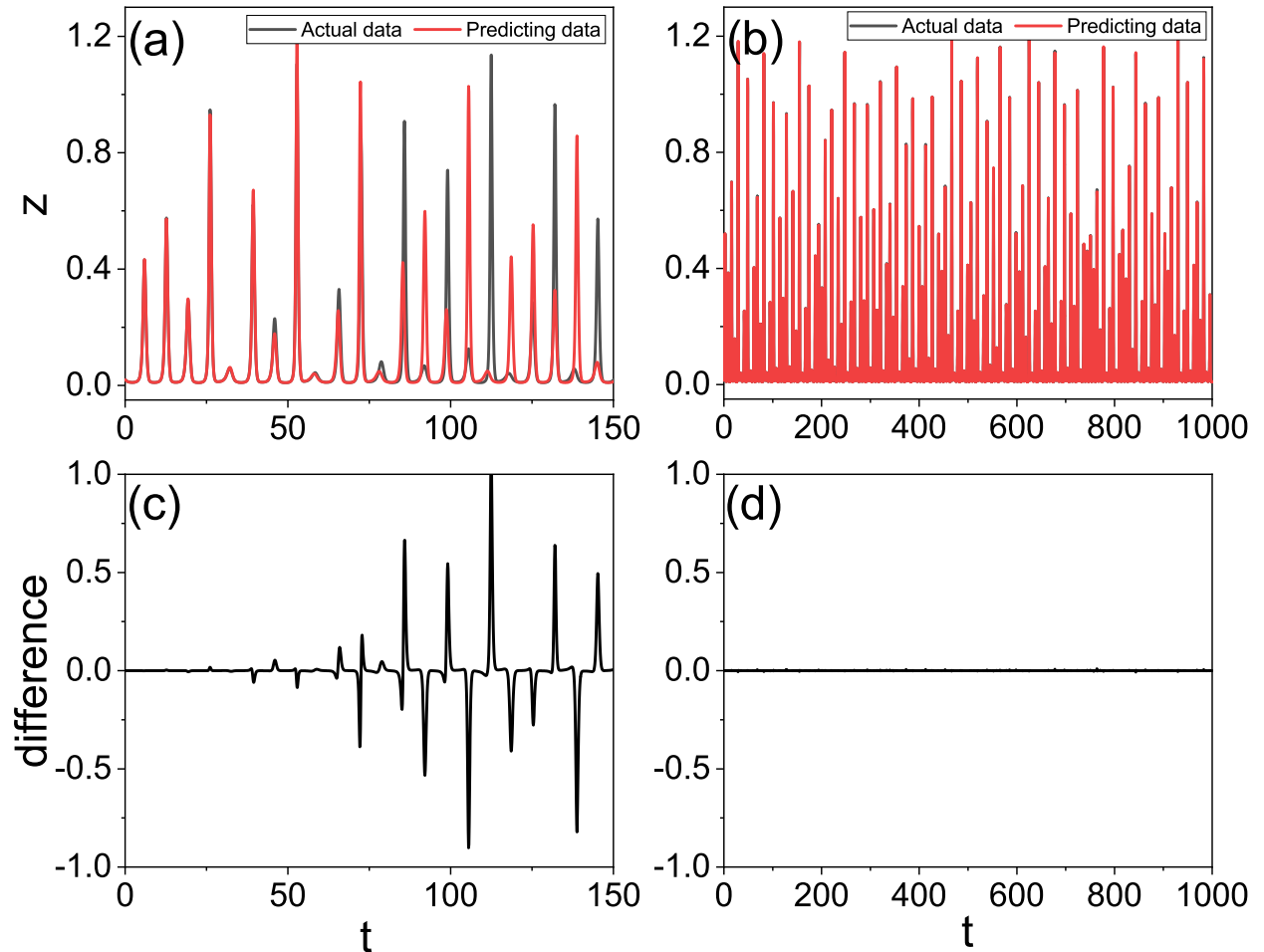

FIG. S4. *Predicting the state evolution of the populations of a chaotic food web.* (a) Predicted and actual state evolution with the conventional reservoir computing scheme without any state updating. (b) Prediction result with rate state updating: once every 50 time steps ($T_0 = 1$ and $T = 50$). The value of the coupling parameter is $c = 1$. (c,d) Evolution of the prediction error corresponding to the cases in (a,b), respectively. With the rare state updating, the prediction error is practically zero in the entire time interval of 10^4 time steps tested.

[1] O. E. Rössler, "Equation for continuous chaos," Phys. Lett. A **57**, 397 (1976).

[2] A. Stefański, P. Perlikowski, and T. Kapitaniak, "Ragged synchronizability of coupled oscillators," Phys. Rev. E **75**, 016210 (2007).

[3] P. Perlikowski, B. Jagiello, A. Stefanski, and T. Kapitaniak, "Experimental observation of ragged synchronizability," Phys. Rev. E **78**, 017203 (2008).

[4] 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).

[5] E. N. Lorenz, "Deterministic nonperiodic flow," J. Atmos. Sci. **20**, 130 (1963).

[6] J. L. Hindmarsh and R. M. Rose, "A model of neuronal bursting using three coupled first order differential equations," Proc. Roy. Soc. London Ser. B Biol. Sci. **221**, 87 (1984).

[7] B. Blasius, A. Huppert, and L. Stone, "Complex dynamics and phase synchronization in spatially extended ecological systems," Nature **399**, 354 (1999).