

Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks: Role of network spectral radius

Junjie Jiang¹ 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 19 August 2019; published 29 October 2019)

A common difficulty in applications of machine learning is the lack of any general principle for guiding the choices of key parameters of the underlying neural network. Focusing on a class of recurrent neural networks—reservoir computing systems, which have recently been exploited for model-free prediction of nonlinear dynamical systems—we uncover a surprising phenomenon: the emergence of an interval in the spectral radius of the neural network in which the prediction error is minimized. In a three-dimensional representation of the error versus the time and spectral radius, the interval corresponds to the bottom region of a “valley.” Such a valley arises for a variety of spatiotemporal dynamical systems described by nonlinear partial differential equations, regardless of the structure and the edge-weight distribution of the underlying reservoir network. We also find that, while the particular location and size of the valley depend on the details of the target system to be predicted, the interval tends to be larger for undirected than for directed networks. The valley phenomenon can be beneficial to the design of optimal reservoir computing, representing a small step forward in understanding these machine-learning systems.

DOI: [10.1103/PhysRevResearch.1.033056](https://doi.org/10.1103/PhysRevResearch.1.033056)

I. INTRODUCTION

Recent years have witnessed a growing interest in exploiting machine-learning algorithms for predicting the state evolution of nonlinear dynamical systems [1–10]. Reservoir computing, a form of echo state [11,12] or liquid state [13] machines that are fundamentally recurrent neural networks, stands out as a viable paradigm for model-free, data-based prediction of chaotic systems [3–6,9,14]. A general reservoir computing scheme consists of an input layer, a reservoir that is a high-dimensional or networked neural dynamical system, and an output layer. The input layer maps the given, low-dimensional time series or sequential data into the high-dimensional phase space of the reservoir network, and the output layer maps the evolution of the high-dimensional dynamical variables of the reservoir back into low-dimensional time series as readout. During the training phase, the output is compared with the original input data from the target system, and parameters of the output layer are tuned to minimize the difference. A properly trained reservoir-computing system without any input is itself a dynamical system whose evolution from a given set of initial conditions represents the prediction of the state evolution of the target system from that particular initial-condition set. Since the high-dimensional neural network system constituting the reservoir is predetermined

and fixed, learning can be accomplished rapidly with high efficiencies and at a low cost. Physically, reservoir computing can be realized electronically with time-delay autonomous Boolean systems [1] or implemented using high-speed photonic devices [2].

There are two types of parameters in reservoir computing or in machine learning in general: a predefined, fixed set of parameters and a set of tunable parameters whose values are determined through the training or learning process. For convenience, we call the former *free parameters* and the latter *learning parameters*. An extremely challenging issue in machine learning is the lack of general rules or criteria for selecting the predefined parameters. The common practice is mostly a random, brute-force type of trial-and-error process to determine the parameter values. Because of the vast complexity of the neural network dynamics associated with machine learning, to develop any general and systematic approach to choosing the predefined parameters has remained an outstanding problem, with little possibility of viable solutions in sight.

In this paper, we report a general phenomenon associated with reservoir computing as applied to model-free and data-based prediction of nonlinear dynamical systems, which can be used to guide the choice of the core component of the neural computing system: the high-dimensional dynamical backbone neural network constituting the reservoir. To be as general as possible, we assume that the reservoir is described by a complex weighted network. Because of the large size of the network, a vastly large number of predefined parameters (and properties) will then need to be determined, such as the network topology, the average degree, the network size, and the edge weights, making any systematic selection of these parameters/properties a practically impossible task. To make our study feasible, we consider both directed and undirected

*ying-cheng.lai@asu.edu

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

topology and fixed the network structure, leaving only the edge weights as the set of free parameters. Even then, the number of possible parameter choices is enormous. Quite surprisingly, we find that, in spite of the large number of free parameters, the one that is key to the success of reservoir computing is the spectral radius of the complex neural network. In particular, we find that there exists an interval of the values of the network spectral radius within which the training error associated with reservoir computing is minimized. In a three-dimensional plot of the error versus the time and spectral radius, a valleylike structure with a flat bottom of finite size with near-zero error emerges. This means that, regardless of the network details, insofar as its spectral radius is chosen from the valley region, model-free prediction with reservoir computing can be guaranteed. We establish this result through a number of nonlinear dynamical systems arising from different physical contexts: spatiotemporal systems described by the nonlinear Schrödinger equation (NLSE), the Kuramoto Sivashinsky equation (KSE), and the one-dimensional (1D) complex Ginzburg-Landau equation (CGLE). Considering that general phenomena for guiding the choices of parameter values are rare in the machine learning literature, our finding is encouraging and may stimulate further efforts in searching for common principles underlying the working of machine learning not only in reservoir computing but also beyond.

II. RESERVOIR COMPUTING

There are two major types of reservoir computing systems: echo state networks (ESNs) [11] and liquid state machines (LSMs) [13]. The architecture of an ESN is one that is associated with supervised learning underlying recurrent neural networks. The basic principle of ESNs is to drive a large neural network of a random or complex topology—the reservoir network—with the input signal. Each neuron in the network generates a nonlinear response signal. Linearly combining all the response signals with a set of trainable parameters yields the output signal. As for ESNs, an LSM is also a random or complex network of neurons, with the difference being that each neuron receives not only an external input signal but also signals from other neurons in the network. The networked system is thus effectively a spatiotemporal nonlinear dynamical system, where trainable, linearly discriminant units are used to map the spatiotemporal patterns of the network into proper output signals. Structurewise, LSMs are more complicated than ESNs.

For simplicity, we focus on ESNs. A schematic of a typical ESN is shown in Fig. 1, where the reservoir computing machine consists of three components: (i) an input subsystem that maps the low-dimensional (say M) input signal into a (high) N -dimensional signal through the weighted $N \times M$ matrix \mathbf{W}_{IR} ; (ii) the reservoir network of N neurons characterized by \mathbf{W}_{res} , a weighted network matrix of dimension $N \times N$; and (iii) an output subsystem that converts the N -dimensional signal from the reservoir network into an L -dimensional signal through the output weighted matrix \mathbf{W}_{RO} , where $L \sim M \ll N$. In Fig. 1, the three components are denoted I, R, and O, respectively.

The working of an ESN can be described as follows. As shown in Fig. 1, the training phase is represented by the blue

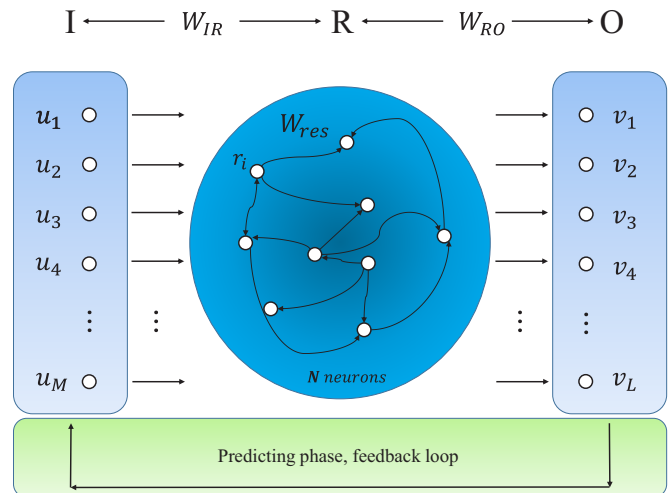


FIG. 1. Basic structure of reservoir computing. The left blue box represents an input subsystem that maps the M -dimensional input data to a vector of much higher dimension N , where $N \gg M$. The blue circle in the middle denotes the reservoir system, which can be, for example, a complex neural network of N interconnected neurons, whose connection structure is characterized by the $N \times N$ weighted matrix \mathbf{W}_{res} . The dynamical state of the i th neuron in the reservoir is r_i . The blue box on the right represents the output module, which converts the N -dimensional state vector of the reservoir network into an L -dimensional output vector, where $N \gg L$. The mapping from the input module to the reservoir is described by the $N \times M$ weighted matrix \mathbf{W}_{IR} , and that from the reservoir to the output module by the $L \times N$ weighted matrix \mathbf{W}_{RO} . During the training phase, the three blue boxes are activated. In this case, the whole computing device is effectively a nonlinear dynamical system with external input. In the prediction phase, the external input is cut off and the output data are directly fed back to the reservoir (green box), so the system is one without any external driving.

blocks. The input multidimensional data have the dimension $M \times N_t$, where M is the dimension of the input data vector $\mathbf{u}(t)$ at time t and N_t is the number of time steps used in the training phase: $t = 0, dt, 2dt, \dots, (N_t - 1)dt$. The input data vector to the reservoir network is $\mathbf{W}_{IR} \cdot \mathbf{u}(t)$. The values of the elements in \mathbf{W}_{IR} are obtained from a uniform distribution in $[-\alpha, \alpha]$. Every neuron in the reservoir receives one component of the input data vector to the reservoir. Typically, the reservoir is a large, sparse, directed or undirected random network with average degree k , which is described by a weighted adjacency matrix \mathbf{W}_{res} , whose largest absolute eigenvalue is the network spectral radius ρ . For a given value of ρ , we choose the values of all the elements of \mathbf{W}_{res} randomly from a uniform distribution $[U(0, 1)]$ and rescale all the values so that its largest eigenvalue is ρ . The state of the whole reservoir at time t is an N -dimensional vector $\mathbf{r}(t)$, where each dimension represents the dynamical state of an individual node. For the i th node, its state is denoted $r_i(t)$. The initial state of the reservoir is $\mathbf{r}(0) = \mathbf{0}$. The state of the reservoir is updated at every time step dt according to

$$\mathbf{r}(t + dt) = f(\mathbf{W}_{res} \cdot \mathbf{r}(t) + \mathbf{W}_{IR} \cdot \mathbf{u}(t)), \quad (1)$$

where f is a function that activates every element in the reservoir state vector, a typical choice of which is the tanh function. After evolving Eq. (1) for all the time steps in \mathbf{u} , we get an $N \times (N_r + 1)$ -dimensional matrix of the reservoir state \mathbf{r} . We disregard the first S steps of the reservoir as transients. Since the activation function tanh is odd, it is necessary [6] to normalize the vector \mathbf{r} by taking the squares of its even elements. This leads to a new state vector \mathbf{r}' . All the training is done with respect to the normalized reservoir state vector \mathbf{r}' and the output vector \mathbf{v} , which updates the output matrix \mathbf{W}_{RO} . The training phase is completed when the output $L \times N$ matrix \mathbf{W}_{RO} has been determined.

The output matrix \mathbf{W}_{RO} can be calculated using the regression scheme that minimizes the loss function,

$$\mathcal{L} = \sum_{t=d+1}^{N_t} \|\mathbf{v}(t) - \mathbf{W}_{RO}\mathbf{r}'(t)\|^2 + \Gamma \|\mathbf{W}_{RO}\|^2, \quad (2)$$

where $\|\mathbf{W}_{RO}\|^2$ is the sum of squared elements of \mathbf{W}_{RO} . The parameter Γ is a small positive regularization constant introduced for preventing overfitting by imposing a penalty on large values of the fitting parameters. The regularized regression can be described as

$$\mathbf{W}_{RO} = \mathbf{v} \cdot \mathbf{r}'^T \cdot (\mathbf{r}' \cdot \mathbf{r}'^T + \Gamma \mathbf{I})^{-1}. \quad (3)$$

There are two types of strategies to set the initial state of the reservoir network. One can simply continue from the training phase to predict the system, i.e., one continues to use the reservoir state at the final time step of the training phase for prediction, entailing a “warm start” of the prediction phase. Alternatively, one can start the prediction from a different data set, where the initial reservoir state is an N -dimensional zero vector. For a number of time steps at the beginning of the prediction phase, one uses the true state \mathbf{u} of the system in Eq. (1) to drive the reservoir to a functioning state. This is essentially a “cold start” strategy.

After a warm or a cold start, the dynamical state of the reservoir has been activated. With the \mathbf{W}_{RO} matrix determined during the training phase, the output of the reservoir is given by

$$\mathbf{v}(t - dt) = \mathbf{W}_{RO} \cdot \mathbf{r}'(t), \quad (4)$$

where $\mathbf{r}'(t)$ has been updated from $\mathbf{r}(t)$ with the elements in the even rows squared. After obtaining $\mathbf{v}(t - dt)$, one replaces $\mathbf{u}(t)$ with $\mathbf{v}(t - dt)$ and the reservoir system can produce the predicting time series continuously. This feedback process is illustrated by the green block in Fig. 1.

In a recent work [6], ESNs have been applied to predicting the dynamical state of the spatially extended KSE in the chaotic regime. It was demonstrated that, with properly chosen parameters, an ESN can predict the dynamical states of the KSE in the entire spatial domain for several Lyapunov time.

III. FUNDAMENTAL ROLE OF THE SPECTRAL RADIUS OF THE RESERVOIR NETWORK IN PREDICTING SPATIOTEMPORAL DYNAMICAL SYSTEMS

We demonstrate that the spectral radius of the reservoir complex network plays a fundamental role in achieving successful prediction. We substantiate this finding through a number of spatiotemporal dynamical systems arising from physics: the NLSE, the KSE, and the CGLE.

A. Nonlinear Schrödinger equation

The NLSE has been a paradigm to study nonlinear wave propagation in fields such as optics and hydrodynamics [15]. Among the analytic solutions of the NLSE are physically significant phenomena such as “breathers,” “solitons,” and rogue waves in a finite background that has been experimentally observed in nonlinear fiber optics [16–18]. To be concrete, we investigate the feasibility of exploiting reservoir computing for predicting finite-background soliton solutions in the NLSE, with a particular eye towards elucidating the role of network spectral radius in the prediction.

The finite-background soliton solutions include Akhmediev breathers, Kuznetsov-Ma solitons, and Peregrine solitons. We consider analytic solutions of the NLSE representing Akhmediev breathers and Kuznetsov-Ma solitons. The dimensionless NLSE reads

$$i \frac{\partial \psi}{\partial x} + \frac{1}{2} \frac{\partial^2 \psi}{\partial t^2} + |\psi|^2 \psi = 0, \quad (5)$$

where the envelope $\psi(x, t)$ is a function of the propagation distance x and comoving time t . The analytic solution of the NLSE describing general modulation instabilities in optics was first obtained in Ref. [19] and is given by

$$\psi(x, t) = e^{ix} \left[1 + \frac{2(1 - 2a) \cosh(bx) + ib \sinh(bx)}{\sqrt{a} \cos(\omega t) - \cosh(bx)} \right], \quad (6)$$

where $b = \sqrt{8a(1 - 2a)}$, $\omega = \sqrt{2(1 - 2a)}$, and the positive parameter a determines the physical properties of the solution. For example, for $a = 0.25$, the solution corresponds to Akhmediev breathers, and for $a = 0.7$, Kuznetsov-Ma solitons arise. To generate the true data from Akhmediev breathers, we set $x \in [-\pi, \pi]$ and discretize the space into 64 lattice points. The spatial step size is $dx = 2\pi/63$ and the time step is $dt = \pi/100$. To generate the true Kuznetsov-Ma soliton data, we set $T \in [-\pi, \pi]$ and employ the same spatial discretization scheme. Note that, physically, Akhmediev breathers and Kuznetsov-Ma solitons are qualitatively similar through an exchange of the time and space variables.

We also test a more complicated type of wave patterns, those generated by the collision of two solitons [20,21], where the corresponding solution can be obtained as the nonlinear superposition of two first-order Akhmediev breathers for $0 < a_1, a_2 < 0.5$,

$$\psi_{12}(x, t) = \psi_0 + \frac{2(l_1^* - l_1)s_1r_1^*}{|r_1|^2 + |s_1|^2} + \frac{2(l_2^* - l_2)s_1r_2^*}{|r_{12}|^2 + |s_{12}|^2}, \quad (7)$$

where $l_1 = i\sqrt{2a_1}$, $l_2 = i\sqrt{2a_2}$, and $*$ represents the complex conjugate. (A complete expression of solution is presented in Appendix D.) To generate the data for the soliton collision wave pattern, we again set $x \in [-\pi, \pi]$ and discretize the

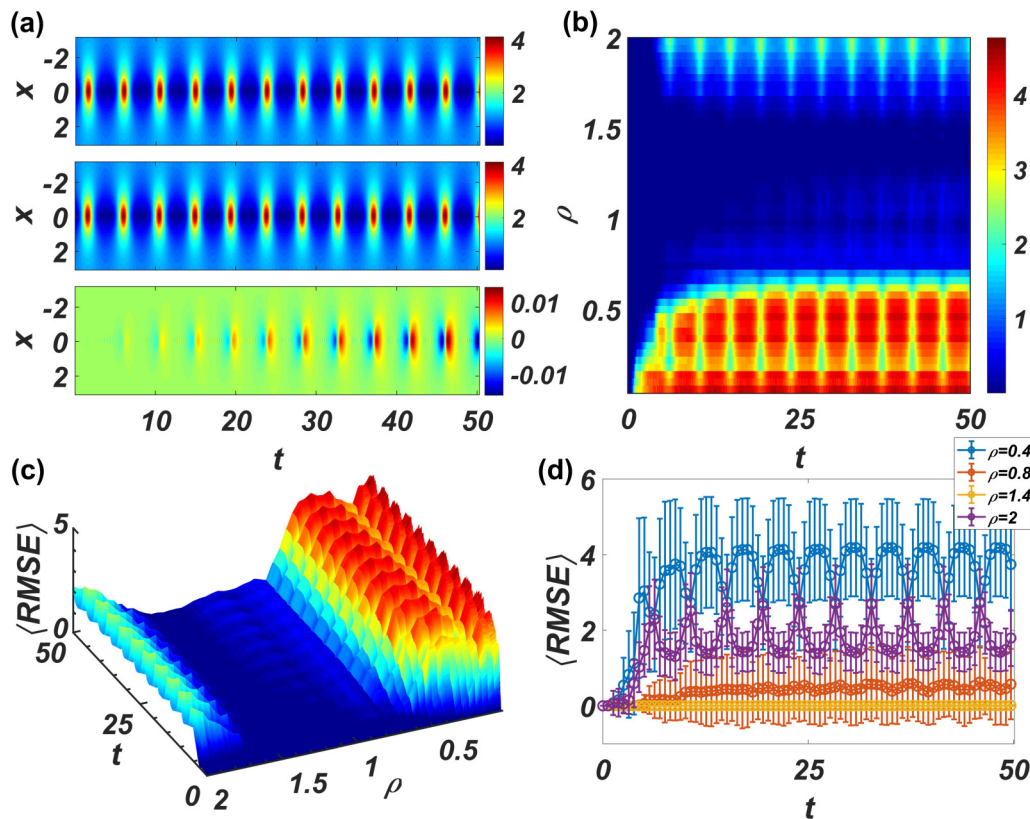


FIG. 2. Using reservoir computing to predict Akhmediev breathers and the emergence of an optimal interval in the spectral radius. (a) For spectral radius $\rho = 1.5$, a successful case of prediction of Akhmediev breathers, where the top panel shows the time evolution of the true solution, the middle panel displays the predicted solution from reservoir computing after training, and the bottom panel depicts the difference between the true and the predicted solutions. The color bar indicates the scale of the spatiotemporal wave magnitude $|\psi(x, t)|$. (b) For $0 < \rho \leq 2$ (ordinate), time evolution of the ensemble-averaged RMSE, denoted $\langle \text{RMSE} \rangle$, where, for each fixed value of ρ , 100 random realizations of the reservoir system are used to calculate the average and the color bar indicates the scale of the $\langle \text{RMSE} \rangle$ values. (c) Three-dimensional view of (b). The emergence of a valley interval in ρ with a minimized prediction error can be seen unequivocally in (b) and (c). (d) Detailed time evolution of $\langle \text{RMSE} \rangle$ for four specific values of ρ with standard deviation. (More quantitative details can be found in Appendixes A and B.) Other parameters of the reservoir-computing system are $\alpha = 1$, $N = 4992$, $k = 3$, $M = L = 64$, $N_t = 8010$, $S = 10$, and $\Gamma = 1 \times 10^{-4}$.

space into 64 points. The time step is $dt = \pi/40$. For illustrative purposes, we consider two parameter settings: $(a_1 = 0.14, a_2 = 0.34)$ and $(a_1 = 0.42, a_2 = 0.18)$.

Figure 2 shows the results of using reservoir computing to predict the spatiotemporal evolution of Akhmediev breathers for $a = 0.25$. The dimension of the input data is 64; so is that of the output data. The number of nodes (neurons) in the reservoir network is chosen to be $N = 4992 = 64 \times 78$, so every dimension of the input data is connected with 78 neurons in the reservoir. We choose $N_t = 8010$ with the transient time $\tau = 10$, so the training phase contains approximately 56 solitons. In the prediction phase, we choose the “warm start” strategy to initiate the dynamical evolution of the reservoir neural network. Figure 2(a) shows the results of predicting 11 Akhmediev breathers over 1600 time steps (corresponding to $t \approx 50$). We see that the occurrence in time of approximately five solitons can be predicted with a relatively small error.

To search for any possible general rule that can lead to prediction success as exemplified in Fig. 2(a), we systematically vary the value of the spectral radius ρ . Extensive tests reveal a remarkable phenomenon: the emergence of an optimal interval of the ρ values in which the prediction error is minimized,

as shown in Fig. 2(b), where the time evolution of the ensemble average of the root mean square error ($\langle \text{RMSE} \rangle$) between true and predicted solutions for $0 < \rho \leq 2$ is displayed. For each fixed ρ value, we generate 100 reservoir systems with random weights for \mathbf{W}_{IR} , where the reservoir network has a random topology with randomly distributed edge weights (so \mathbf{W}_{res} is effectively a random matrix), and calculate $\langle \text{RMSE} \rangle$. Figure 2(c) is a three-dimensional representation of Fig. 2(b), where the existence of the optimal ρ interval with minimum error can be identified: $\rho \in [0.8, 1.6]$. In fact, for $\rho < 0.7$, the values of the error $\langle \text{RMSE} \rangle$ are dramatically large in comparison with those in the valley. As the value of ρ is increased from about 1.6, the error value grows approximately linearly. Figure 2(d) shows the detailed time evolution of the error $\langle \text{RMSE} \rangle$ (with standard deviation) for $\rho = 0.4, 0.8, 1.4$, and 2, where the case of $\rho = 1.4$ (in the valley) has near-zero values of $\langle \text{RMSE} \rangle$ as well as near-zero standard deviations. The results in Fig. 2 thus indicate that, insofar as the spectral radius of the random reservoir network is chosen from the valley, the reservoir system performs well for predicting Akhmediev breathers, regardless of the network structure and edge-weight distribution. (More detailed information about the behavior of RMSE in this case can be found in Appendixes A and B.)

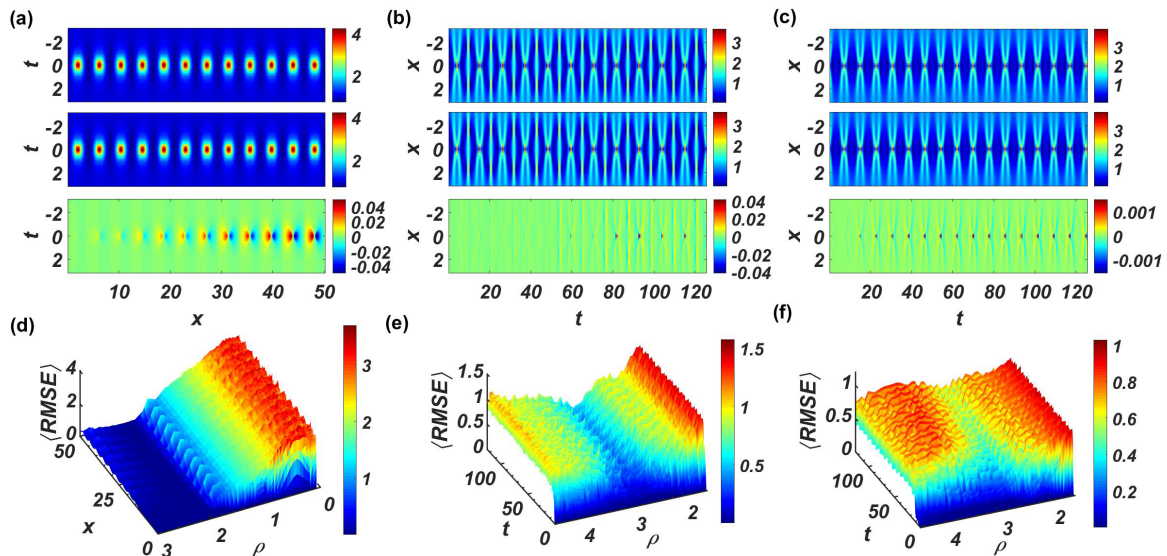


FIG. 3. Emergence of a valley in the prediction error versus the spectral radius of the reservoir network for predicting Kuznetsov-Ma solitons and soliton collision in the NLSE. (a)–(c) True (upper panel) and predicted (middle panel) wave patterns as well as the difference (lower panel) in wave function for Kuznetsov-Ma solitons: soliton collision for $(a_1 = 0.14, a_2 = 0.34)$ and $(a_1 = 0.42, a_2 = 0.18)$, respectively. (d)–(f) Corresponding time evolution of the ensemble-averaged prediction error $\langle \text{RMSE} \rangle$ for systematically varying ρ values. For each fixed ρ value, 100 random reservoir systems are used to calculate the average error and the color bar indicates the scale of the $\langle \text{RMSE} \rangle$ value. For predicting the Kuznetsov-Ma solitons, if the value of ρ is chosen from the valley, then all realizations lead to near-zero errors (d). However, for the case of predicting soliton collision, only about half of the realizations yield near-zero errors even when the value of ρ is chosen from the valley (outside the valley, large errors arise for nearly all realizations). Other parameter values are $N = 4992, k = 3, M = L = 64, N_t = 8010, S = 10, \Gamma = 1 \times 10^{-4}, \alpha = 1.0$ for (a) and (d) and $\alpha = 3.0$ for (b), (c), (e), and (f). Information about the standard deviation of $\langle \text{RMSE} \rangle$ can be found in Appendixes A and B.

The existence of an optimal interval in the spectral radius of the reservoir network also holds for Kuznetsov-Ma solitons and colliding solitons. Specifically, Figs. 3(a) and 3(d) show that a properly designed reservoir computing system can predict the solutions of the NLSE in the regime of Kuznetsov-Ma solitons, where Fig. 3(d) reveals that the ensemble-averaged error $\langle \text{RMSE} \rangle$ is minimized for $\rho \in [1.98, 2.34]$ —the valley. [Note that the ρ value of the reservoir network in Fig. 3(a) is 1 with the minimum error in Fig. 3(d).] As the value of ρ is decreased from the valley interval, the error on $\langle \text{RMSE} \rangle$ increases rapidly. However, $\langle \text{RMSE} \rangle$ tends to increase slowly when the value of ρ is larger than the valley interval. While the behaviors are similar to those in the prediction of Akhmediev breathers, the locations of the valley interval for the two cases are different, where for the regime of Kuznetsov-Ma solitons, the valley occurs in a relatively larger interval of ρ values.

The results from predicting the patterns of colliding solitons are shown in Figs. 3(b) and 3(c). While reservoir computing is able to predict wave patterns, in this case there is no guarantee that, for every random reservoir network with a ρ value taken from the valley indicated in Figs. 3(e) and 3(f), prediction can be successful. For example, for $\rho = 2.88$, about 50% of the cases can yield a good prediction result. However, if the value of ρ is not chosen from the valley interval, reservoir computing fails to make any meaningful prediction. A similar behavior has been found for a different case of colliding solitons. We note that, when predicting Akhmediev breathers and Kuznetsov-Ma solitons, successful prediction can be achieved for any random network whose value of the spectral radius lies in the valley. The case of

colliding solitons in the NLSE is thus more unpredictable than that of Akhmediev breathers and Kuznetsov-Ma solitons. Nonetheless, in spite of the difficulty, the existence of a valley interval leading to optimal prediction performance also holds for the case of colliding solitons.

The solution of soliton collision in the NLSE represents a difficult case where, as shown in Fig. 3, even for the optimal value of the spectral radius, only about 50 of 100 ensemble realizations lead to acceptable prediction results in terms of both accuracy and time. The main reason is that the process of soliton collision necessarily involves a possible change in the “climate” of the dynamical state, as the two solitons can bounce back from each other or merge [22]. In the case of bouncing back, the feature or climate of the dynamical state of the system remains unchanged before and after the collision. In this case, the reservoir computing system is able to make accurate predictions. In the latter case of merging, the state climate has changed completely before and after the collision, rendering inaccurate predictions, as the neural network was mostly trained in the presence of two solitons.

B. Predicting spatiotemporal chaotic solutions of the Kuramoto-Sivashinsky equation

We test reservoir-computing-based prediction of chaotic solutions of the KSE (the original system that was used to demonstrate the power of reservoir computing to predict spatiotemporal chaotic systems [6]) and show the emergence of a valley interval in the spectral radius of the reservoir network with minimum prediction error. The KSE is a 1D

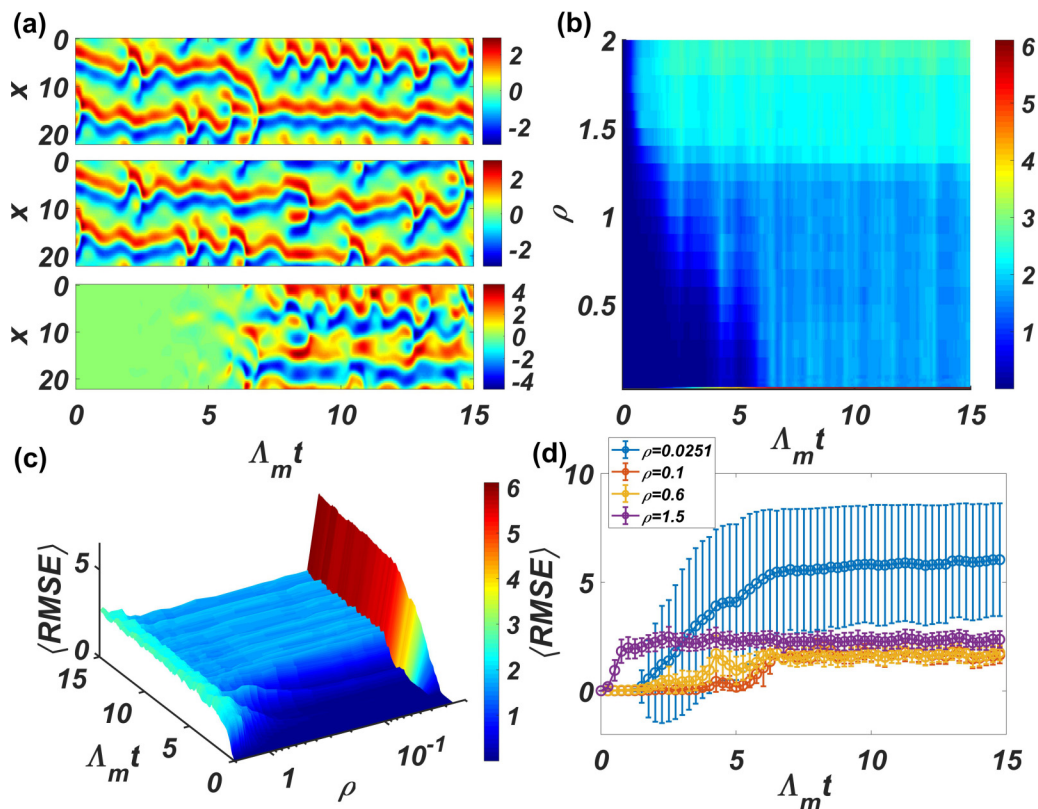


FIG. 4. Emergence of a valley interval in the spectral radius of the reservoir network with minimum error for predicting spatiotemporal chaotic solutions of the KSE. (a) An example of successful prediction for $\rho = 0.1$. Top panel: True spatiotemporal evolution of a typical chaotic solution of the KSE. Middle panel: Predicted spatiotemporal evolution. Lower panel: Difference between the true and the predicted solutions (true minus predicted). The color bar indicates the scale of $u(x, t)$. (b) Time evolution of ensemble-averaged RMSE (over 100 random realizations of the reservoir system) for systematically varying ρ values in the interval $(0.0, 2.0]$, where the color bar indicates the scale of $\langle \text{RMSE} \rangle$. Note that the evolution time is represented as $\Lambda_m t$, where $\Lambda_m \approx 0.05$ is the maximum Lyapunov exponent of the chaotic solution and one unit of $\Lambda_m t$ corresponds to one Lyapunov time. (c) Three-dimensional representation of $\langle \text{RMSE} \rangle$ in the $(\rho, \Lambda_m t)$ plane, revealing the emergence of a valley interval in ρ in which $\langle \text{RMSE} \rangle$ is minimized. (d) Time evolution of $\langle \text{RMSE} \rangle$ for four representative values of ρ with standard deviations (more details are given in Appendix A). Among the four cases, the best prediction result is achieved for $\rho = 0.1$, where the chaotic solution can be predicted with near-zero error for about five Lyapunov time. Other parameters of the reservoir computing system are $\alpha = 1$, $N = 4992$, $k = 3$, $M = L = 64$, $N_i = 70010$, $S = 10$, and $\Gamma = 1 \times 10^{-4}$.

nonlinear PDE given by

$$\frac{\partial u}{\partial t} = -u \frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4}, \quad (8)$$

where $u(x, t)$ is a scalar field. We set the 1D spatial domain to be $x \in [0, 22]$. To obtain the true solution $u(x, t)$, we divide the spatial domain evenly using 64 grid points and numerically solve the KSE with time step $dt = 0.25$. We thus have 64 time series, one from each grid point. Due to the chaotic nature of the KSE, even with reservoir computing it is not possible to predict the behavior of $u(x, t)$ for a relatively long time, and the demonstrated prediction horizon is a few Lyapunov time [6], defined as $\Lambda_m t$, with Λ_m being the largest Lyapunov exponent of the chaotic solution. An example of successful prediction for about five Lyapunov time is shown in Fig. 4(a), where the value of RMSE is smaller than 0.5. Our main point is that, as for the case of the NLSE, a valley interval in the spectral radius of the reservoir network with minimum error emerges for the chaotic solution of the KSE, as illustrated in Figs. 4(b) and 4(c), where the interval is $0.02 \lesssim \rho \lesssim 0.25$. For $\rho < 0.02$, the ensemble-averaged

prediction error $\langle \text{RMSE} \rangle$ is significantly larger, as shown in Fig. 4(c). As the value of ρ is increased from about 0.25, the prediction horizon decreases dramatically, as shown in Fig. 4(b). The time evolution behaviors of $\langle \text{RMSE} \rangle$ for four representative values of ρ are shown in Fig. 4(d). We see that, for $\rho = 0.0251$, the errors are much larger than those in the other three cases. Thus, in spite of the chaotic nature of the solution of the KSE, the valley phenomenon associated with reservoir-computing-based prediction still occurs, as for the regular solutions of the NLSE.

C. Predicting spatiotemporal chaotic solutions of the complex Ginzburg-Landau equation

The CGLE is a general model for gaining insights into a variety of physical phenomena such as nonlinear waves, chemical reactions, superconductivity, superfluidity, Bose-Einstein condensation, and liquid crystals [23–25]. The equation can generate solutions corresponding to complex physical phenomena such as phase chaos, defect chaos, and coexistence of chaos and the plane wave solution. In a system described by

the CGLE, instabilities lead to the formation of a weakly interacting and incoherent background of low-amplitude waves which, under certain conditions, can collapse locally to generate a high-amplitude event. Because of this feature, the CGLE has been used in previous studies to characterize the statistical properties of the extreme events in spatiotemporal dynamical systems [26] and to articulate control strategies [27,28].

In one spatial dimension, the CGLE is written as

$$\frac{\partial u}{\partial t} = u + (1 + i\alpha) \frac{\partial^2 u}{\partial x^2} - (1 + i\beta)|u|^2 u, \quad (9)$$

where $u(x, t)$ is a complex function of space x and time t , and α and β are parameters characterizing linear and nonlinear dispersion, respectively. To be concrete, we focus on the parameter region of defect chaos [23], e.g., $\alpha = 2$ and $\beta = -2$. We set the spatial domain to be $x \in [-9, 9]$. To generate the data for the reservoir system and take into consideration the dynamical complexity of the solutions of the CGLE, we solve Eq. (9) numerically using the pseudospectral and exponential-time differencing scheme [29], where the spatial domain is divided uniformly into 32 subregions and the integration time step is $dt = 0.0001$. From the numerical solutions, we perform time-domain sampling with $dt = 0.07$.

Because of the complex nature of the scalar field $u(x, t)$, two separate input-data streams to the reservoir system are necessary, corresponding to the real and imaginary parts of $u(x, t)$, respectively. (We have verified that the reservoir system fails to produce any meaningful prediction if the module $|u(x, t)|$ is used as the input data.) Figures 5(a) and 5(b) show an example of successful prediction over a horizon of about four Lyapunov time, where the spectral radius value is $\rho = 0.1$. The existence of an optimal valley interval in ρ guaranteeing a similar prediction performance is shown in Figs. 5(c) and 5(d): $0.1 \lesssim \rho \lesssim 2.5$. When the value of ρ is decreased from the left end of the interval, the ensemble-averaged prediction error (RMSE) increases dramatically. Likewise, when ρ is increased from the right end of the interval (e.g., from 3.0 to 4.0), the predicted time with (RMSE) less than about 0.5 decreases monotonically, as shown in Fig. 5(c). Figure 5(e) presents the behaviors of the time evolution of (RMSE) for five specific values of ρ . We see that in the two cases where the values of the spectral radius are outside the valley interval (i.e., $\rho = 3.9811 \times 10^{-4}$ and $\rho = 3.8$), large prediction errors arise. In fact, the standard deviation associated with the evolution is so large that a scale change in the vertical axis is necessary, as done in Fig. 5(f). For other values of the spectral radius, the standard deviation associated with (RMSE) is small when its value is less than about 0.5. When the value of (RMSE) becomes large and plateaued, the values of the standard deviation are approximately uniform. The values of (RMSE) and its standard deviation for $\rho = 1 \times 10^{-2}$ and $\rho = 2.5$ are somewhat similar, but those for the case of $\rho = 1.1$ are somewhat larger. In spite of the diverse behaviors of (RMSE) and its standard deviation, the valley phenomenon giving rise to an optimal interval in the network spectral radius that minimizes the prediction error holds also for the 1D CGLE, indicating generality of the phenomenon.

IV. EFFECT OF THE RESERVOIR NETWORK STRUCTURE ON PREDICTION

The random reservoir networks employed in the various examples in Sec. III all have directed edges. Will the existence of an optimal valley interval in spectral radius persist if the links in the reservoir network become undirected? To address this question, we consider two types of undirected complex networks, random and small-world networks, and test the prediction performance for Akhmediev breathers in the NLSE. Figures 6(a) and 6(b) show the time evolution of the ensemble-averaged prediction error for systematically varying values of ρ for undirected random and small-world networks, respectively. A quite sizable valley interval in ρ with minimum prediction error arises in each case. In fact, in comparison with the directed network structure, the undirected topology leads to a wider valley interval [e.g., comparing Fig. 6(a) with Fig. 2(c)]. A comparison between Figs. 6(a) and 6(b) indicates that the valley interval for the random network structure is slightly larger than that for the small-world topology. In general, whether the network structure is directed or undirected not only affects the size of the valley interval, but also leads to a different “best” value of the spectral radius for which an absolute minimum in the prediction error can be achieved. We have tested other dynamical patterns in the NLSE as well as the KSE and the CGLE and found that the existence of the best spectral radius region is robust, regardless of whether the edges in the reservoir network are directed or undirected.

V. ERROR IN TRAINING OUTPUT DATA

To gain insights into the behavior of the ensemble-averaged RMSE in prediction, we examine the error associated with the training phase. From Eq. (1), we define the time-averaged error during the training phase as $E = |\mathbf{W}_{\text{RO}} \cdot \mathbf{r}' - \mathbf{v}|$, which measures the difference between the generated and the true training output state vector of the neural network, i.e., the error in one time step after training. Figure 7(a) shows the time-averaged error ($\langle E \rangle$) versus the spectral radius ρ for the KSE with a directed network structure, which exhibits a nonmonotonic behavior. Note that the value of ρ minimizing the error is within the valley interval in Fig. 4(b). The increase in the error away from the minimum value as ρ is increased corresponds to the decrease in the prediction horizon in Fig. 4(b). However, the behavior of ($\langle E \rangle$) as ρ is decreased from the optimal value does not appear to explain the dramatic increase in the ensemble-averaged RMSE in prediction in Fig. 4(c). Figure 7(b) shows a similar behavior of ($\langle E \rangle$) but for the case where the complex neural network has an undirected topology. At present, the behaviors of error growth on the two sides of the valley are not analytically understood.

Figure 7 offers insights into the source of prediction error with implications for the prediction time that reservoir computing can possibly achieve. In Fig. 7, we see that the smallest average prediction error for each step is about 6×10^{-5} for the KSE. For the spatiotemporal chaotic solution of the KSE, the maximum Lyapunov exponent is $\Lambda_m \approx 0.05$. With time step $dt = 0.25$, in the predicting phase, the error will grow to about 0.5 in five Lyapunov time. The prediction time is thus mainly

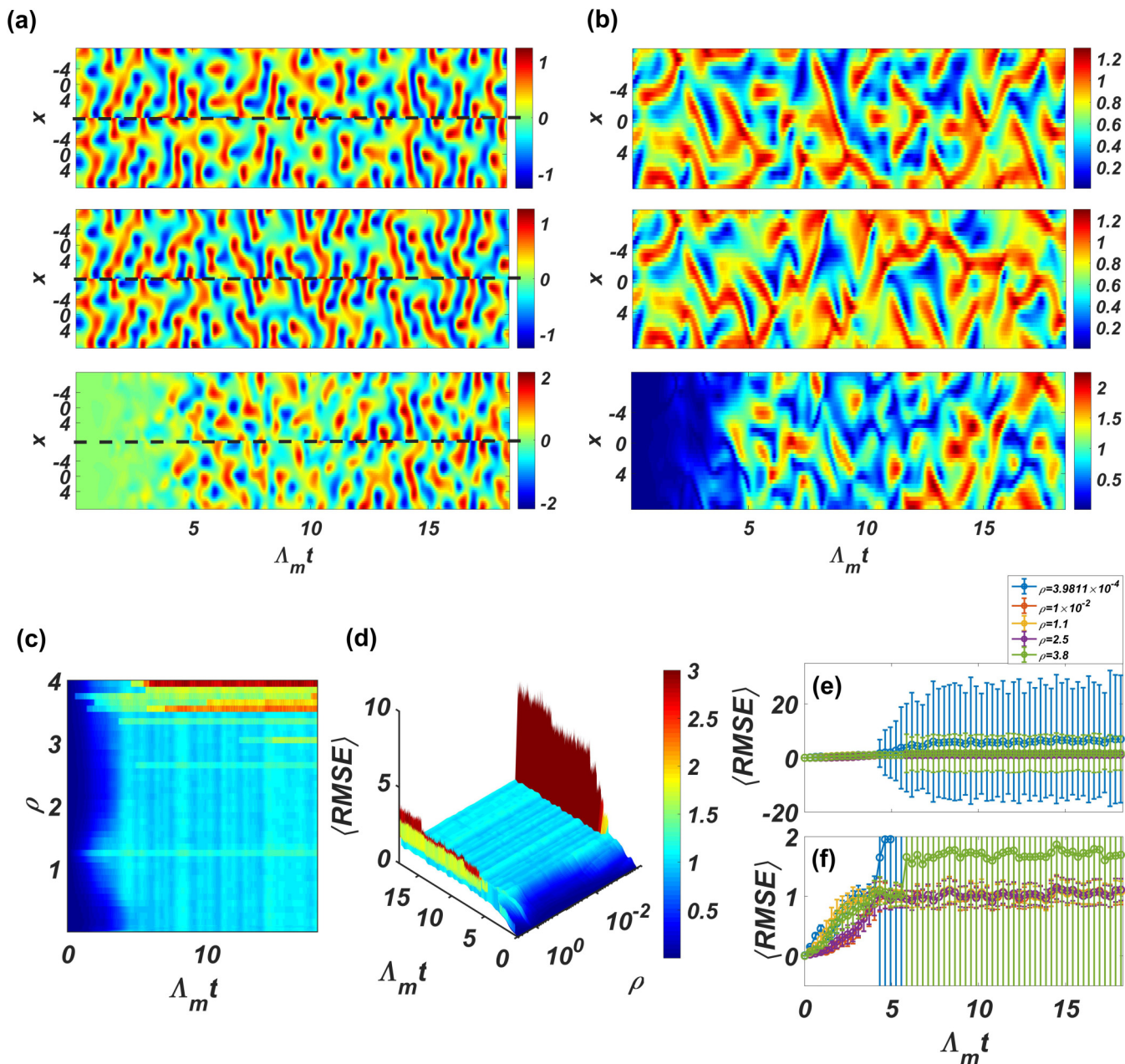


FIG. 5. Emergence of a valley interval in the spectral radius of the reservoir network with minimum error for predicting spatiotemporal chaotic solutions of the CGLE. (a), (b) Example of successful prediction of the spatiotemporal chaotic solution of the one-dimensional CGLE, for which the maximum Lyapunov exponent is $\Lambda_m \approx 0.22$. In (a), the true and predicted real and imaginary parts (above and below the black horizontal dashed lines, respectively) of the spatiotemporal evolution of the solution, together with their difference, are shown. In (b), the true and predicted magnitudes of the complex solution as well as their difference are displayed. The color bars in (a) indicate the scale of the real and imaginary parts of $u(x, t)$. For both (a) and (b), the value of the spectral radius of the reservoir network is $\rho = 0.1$. (c) Ensemble-averaged RMSE calculated from the magnitude value of the complex solution versus ρ and the Lyapunov time, where 100 random reservoir systems are used for each fixed ρ value. (d) Three-dimensional view of $\langle \text{RMSE} \rangle$, where the color bar indicates its scale with the cutoff value of 3.0. The existence of a valley interval in ρ that minimizes the prediction error is unequivocal. (e), (f) Time evolution of $\langle \text{RMSE} \rangle$ (with standard deviation) for five ρ values. (Details of the statistical behavior of $\langle \text{RMSE} \rangle$ are presented in Appendixes A and C.) Other parameters are $\alpha = 1$, $N = 9984$, $k = 3$, $M = L = 64$, $N_t = 80010$, $S = 10$, and $\Gamma = 2 \times 10^{-5}$.

determined by the prediction error of reservoir computing at each time step. While the single-step prediction error can be reduced to a certain extent by fine-tuning the parameters of the neural network, this reduction is often incremental and there is no general method at present to drastically reduce the single-step error.

VI. DISCUSSION

Reservoir computing, a class of recurrent neural networks articulated nearly two decades ago [12–14] for data-based prediction of nonlinear dynamical systems, has recently gained momentum [1–10] as stimulated by the significant growth

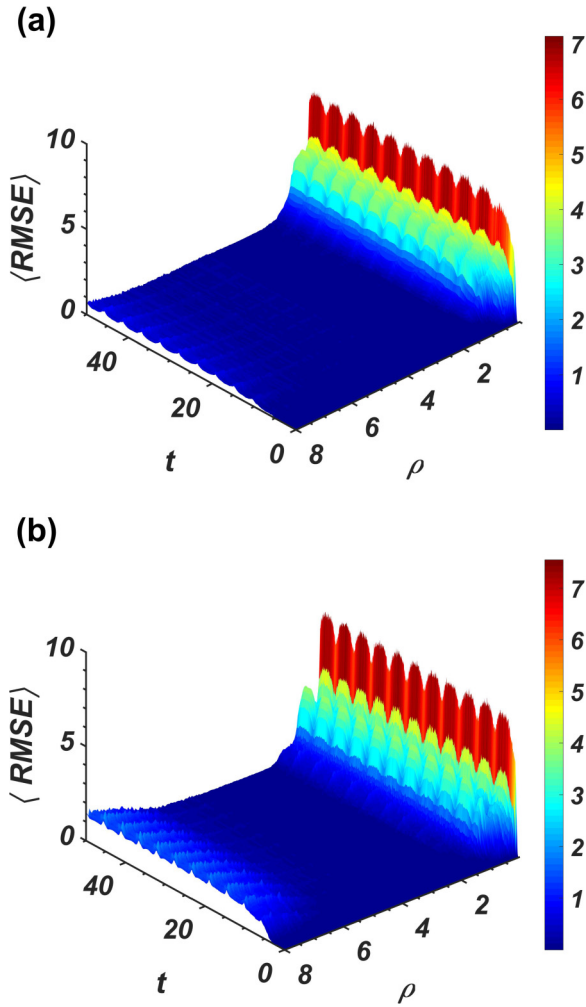


FIG. 6. Emergence of the optimal valley interval in the spectral radius with minimum prediction error for undirected random reservoir networks. (a) For a undirected random reservoir network, time evolution of the ensemble-averaged RMSE (over 100 random reservoirs) for systematically varying values of the network spectral radius for prediction of Akhmediev breathers in the NLSE. (b) Similar plot but for undirected small-world reservoir networks, where the value of the rewiring probability for generating the small-world topology is 0.3. In both cases, a similar valley region arises, indicating that the link topology of the reservoir network, directed or undirected, has little effect on the emergence of the valley. The location and size of the valley interval, however, do depend on the link topology, where undirected networks tend to lead to a larger interval. Parameter values are $\alpha = 1, N = 4992, M = L = 64, N_t = 8010, S = 10, \Gamma = 1 \times 10^{-4}, k = 3$ for (a) and $k = 4$ for (b).

of interest and tremendous advances in modern machine learning. For chaotic dynamical systems, traditional methods [30–33] based on delay-coordinate embedding [34] can usually make short-term predictions, e.g., for about one Lyapunov time. Another prediction framework is based on sparse optimization such as compressive sensing [35,36], but this approach requires that the system’s equations contain mathematically simple terms and time series data from all variables of the system are available. Reservoir-computing-based prediction is model-free and solely data based, and it can

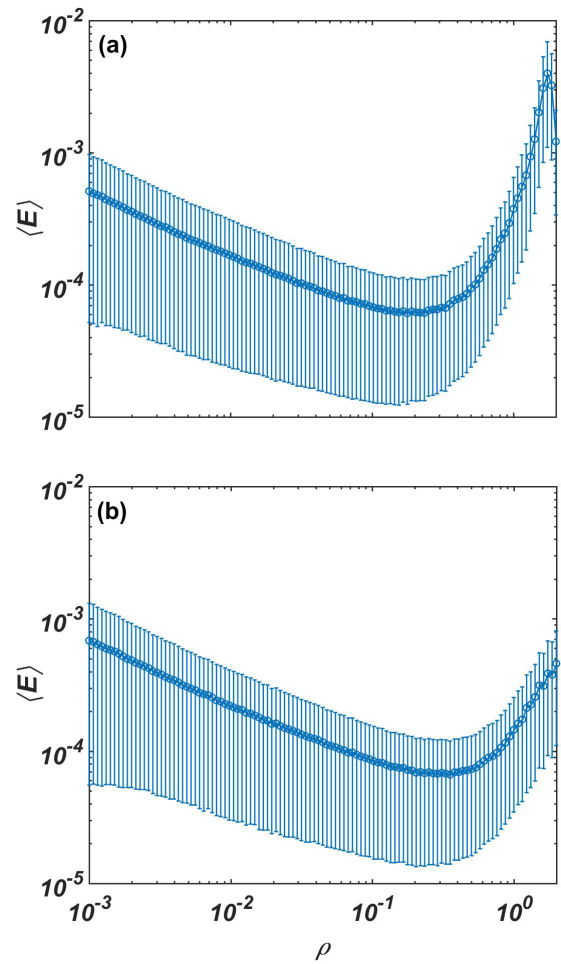


FIG. 7. Behavior of error during the training phase. Time-averaged error $\langle E \rangle$ versus spectral radius ρ for a random reservoir network with (a) a directed topology and (b) an undirected topology with the same average degree as in (a). In each case, the network structure is fixed but the link weights are adjusted to result in systematic variations in the network spectral radius. In both cases, a region of small errors arises, indicating the existence of an optimal interval of spectral radius after training. The location and size of the region are similar to the valley interval in, e.g., Fig. 6. Parameter values are $\alpha = 1, N = 4992, k = 3, M = L = 64, N_t = 70010, S = 10$, and $\Gamma = 1 \times 10^{-4}$.

extend the horizon to about a half-dozen Lyapunov time. This is quite remarkable, defying the conventional wisdom that long-term prediction of the state evolution of a chaotic system is ruled out due to the hallmark of chaos: sensitive dependence on initial conditions. A reservoir computing system, being fundamentally a large neural network, has a large number of parameters whose values need to be fixed. While the values of a subset of parameters can be determined through training with available data, there are still many “free” parameters whose values need to be preset. At the present, for reservoir computing (or for machine learning), there are no general rules that one can rely on to guide the choices of these parameters. Due to the vast complexity and nonlinear structure of reservoir computing systems, to develop mathematical or physical theories to guide systematic choices of the values

of free parameters represents an outstanding and formidably challenging problem, with no hope for solutions in sight.

To make progress, we focus on a spectral property of the reservoir network that typically possesses a complex topology (e.g., random or small world): the spectral radius. Such a network is typically weighted with heterogeneous weights distributed on the set of edges. With variations in the detailed connecting topology and link weights, for the network alone, the parameter space is vast. To make the exploration feasible, we fix the connection topology and assume that only the link weights can vary freely. Even then, combing through all possible parameter variations is a computationally prohibitive task. We thus focus on one question: Is there a range of the spectral radius value that can lead to optimal performance in the sense of minimum prediction error? Note that, with a fixed value of the spectral radius, there are still an infinite number of sets of link weights. Computations with three representative classes of spatiotemporal nonlinear dynamical systems (the NLSE, the KSE, and the CGLE) reveal a remarkable phenomenon: in all cases there exists an optimal interval in the spectral radius that leads to minimum error. (In the three-dimensional plot of the ensemble-averaged prediction error versus the spectral radius and time, the interval appears as a “valley.”) The existence of such a valley interval holds generally true for different systems, regardless of the structure of the complex reservoir network, e.g., directed or undirected, random or small world. Computationally, we find that the interval tends to be larger for undirected than for directed networks. While the finding is purely numerical with no analytic insights, the phenomenon is general and can be exploited for designing optimal reservoir computing systems, representing a small step forward in the study of these machine learning systems.

At present, we do not yet have an analytic understanding of why the value of the spectral radius ρ of the reservoir network needs to be in a certain interval for the neural network system to be effective for prediction. Nonetheless, a heuristic understanding may be attempted. In order for the reservoir system to possess certain predictive power, it must capture the “dynamical climate or complexity” of the target nonlinear system through training. That is, the reservoir system must produce state evolution whose complexity somehow matches that of the target system. In our setting, the network topology is fixed and the variations in the spectral radius are the result of adjusting the edge weights. If the spectral radius is too small, the edge weights are small and the network may be so weakly connected that its collective dynamics are too incoherent to match that of the target system. However, if the spectral radius is too large, the nodal connections in the reservoir network are so tight that the collective dynamics are too coherent, depriving the reservoir computing system of its ability to capture the “climate” of the state evolution of the target system. As a balance of these factors, it is reasonable that, given training data from the dynamical evolution of a specific target system, in general an interval in ρ should emerge in which an optimal match between the complexities of the two systems is achieved. The particular location and size of the interval would depend on the details of the target system to be predicted.

Our work has raised more open questions. For example, a previous work demonstrated that the echo state property

of reservoir computing can be ensured for $\rho < 1$ with zero input but, for non-zero input, the value of ρ can be extended to being larger than 1 [37]. Our study has revealed that, for both the NLSE and the 1D CGLE, the optimal interval in the spectral radius is located in the region $\rho > 1$. Another previous speculation was to regard the spectral radius as a kind of measure of the reservoir’s memory length of the input signal. Consequently, if the input signals are more random and require a larger memory for storage, one should employ a reservoir network with a larger spectral radius for prediction [37,38]. However, our results do not support this point of view. For example, for the NLSE, the dynamical patterns studied are periodic either in space or in time and are thus mostly regular with a minimum degree of randomness, and yet the optimal valley intervals of ρ can be quite different. Why do patterns of similar regularity require different spectral-radius values to be predicted? For the CGLE, in spite of the randomness and complexity of its dynamical evolution, the valley interval is relatively more extensive, from near-zero values to some values far beyond 1. Why can the quite random and complex patterns of the CGLE be predicted with reservoirs of either long or short memory capacity?

ACKNOWLEDGMENTS

This work was supported by the Pentagon Vannevar Bush Faculty Fellowship program sponsored by the Basic Research Office of the Assistant Secretary of Defense for Research and Engineering and funded by the Office of Naval Research through Grant No. N00014-16-1-2828.

APPENDIX A: STANDARD DEVIATION OF THE PREDICTION ERROR

In the text, we have presented the ensemble-averaged error (RMSE) versus the spectral radius of the reservoir network and time, which arises from predicting various states of three types of spatiotemporal dynamical systems. Here we show the standard deviation associated with the error, as in Figs. 8(a)–8(d) for the corresponding cases. In particular, in Figs. 8(a) and 8(b), where the target states are Akhmediev breathers and Kuznetsov-Ma solitons, respectively, the values of the standard deviation are small in the valley region but increase as the value of ρ moves out of the valley, indicating that stable prediction performance can be achieved when choosing the value of ρ in the valley. In Figs. 8(c) and 8(d), where the dynamical states are two distinct cases of soliton collision, the standard deviation is large for all values of ρ tested.

Results of the standard deviation for the KSE and 1D CGLE are shown in Fig. 9, where the dynamical states to be predicted are spatiotemporally chaotic. Again, we observe that the standard deviation associated with the ensemble-averaged error is markedly smaller in the valley interval in the network spectral radius than outside the interval.

APPENDIX B: EXAMPLE OF LONG-TERM PREDICTION OF AKHMEDEV BREATHERS IN THE NLSE

For the dynamical state of Akhmediev breathers in the NLSE, for properly chosen values of the spectral radius, the

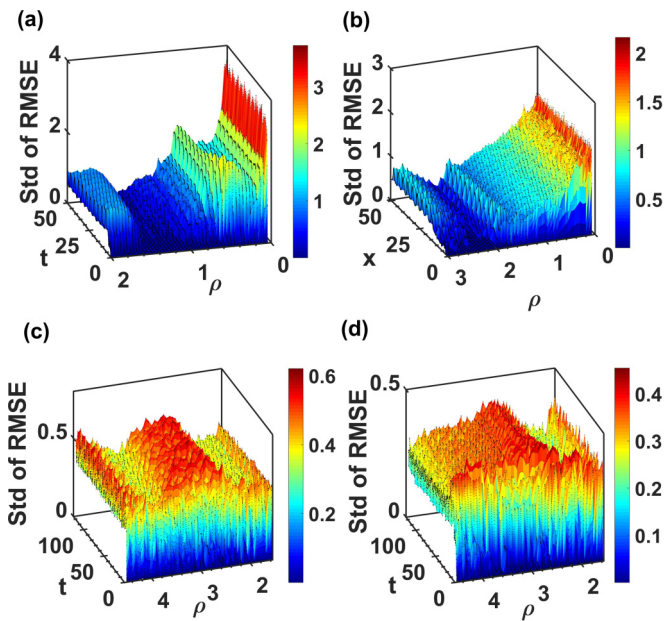


FIG. 8. Standard deviation associated with the ensemble-averaged error in predicting different dynamical states of the NLSE. For each fixed value of the spectral radius ρ , 100 random realizations of the network are used to calculate the standard deviation of the ensemble-averaged prediction error. Shown are the 3D representation of the standard deviation versus the time and ρ for four distinct dynamical states of the NLSE: (a) Akhmediev breathers, (b) Kuznetsov-Ma solitons, (c) a soliton-collision state for $a_1 = 0.14$ and $a_2 = 0.34$, and (d) another soliton-collision state, for $a_1 = 0.42$ and $a_2 = 0.18$.

reservoir computing system is able to make accurate long-term predictions. An example is shown in Fig. 10.

APPENDIX C: ORIGIN OF THE STANDARD DEVIATION IN THE ENSEMBLE-AVERAGED PREDICTION ERROR

The concept of the valley interval discussed in the text is defined with respect to the ensemble-averaged prediction error. That is, for any fixed value of the spectral radius, 100 realizations of the reservoir network are used to calculate the mean error and the standard deviation. In fact, over the different realizations, the prediction error can exhibit quite large variations, even when the value of the spectral radius is

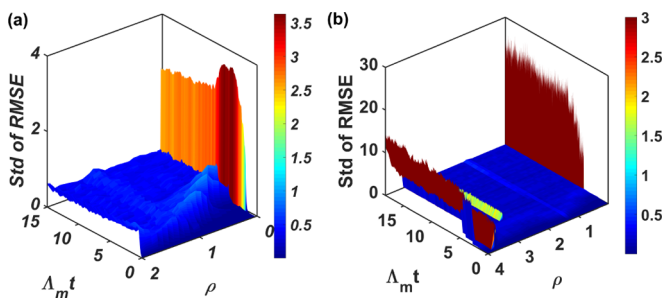


FIG. 9. Standard deviation associated with ensemble-averaged prediction error for (a) the KSE and (b) the 1D CGLE. Legends are the same as in Fig. 8.

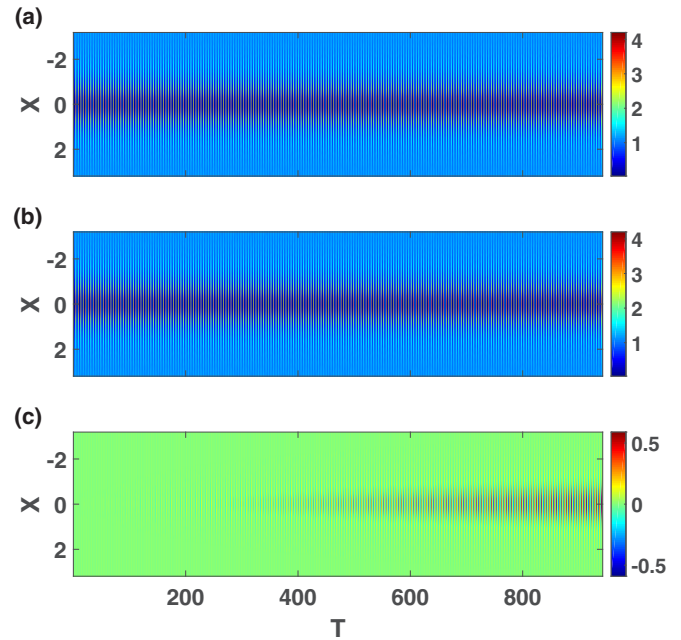


FIG. 10. Example of a long-term prediction of Akhmediev breathers in the NLSE. (a) True spatiotemporal evolution pattern, (b) reservoir-computing predicted pattern, and (c) difference in the instantaneous state between the true and the predicted patterns.

within the valley. Several examples of predicting Akhmediev breathers in the NLSE are shown in Fig. 11, where the error evolution for different realizations (ordinate) is shown for four values of the spectral radius [Figs. 11(a)–11(d)]. For the two

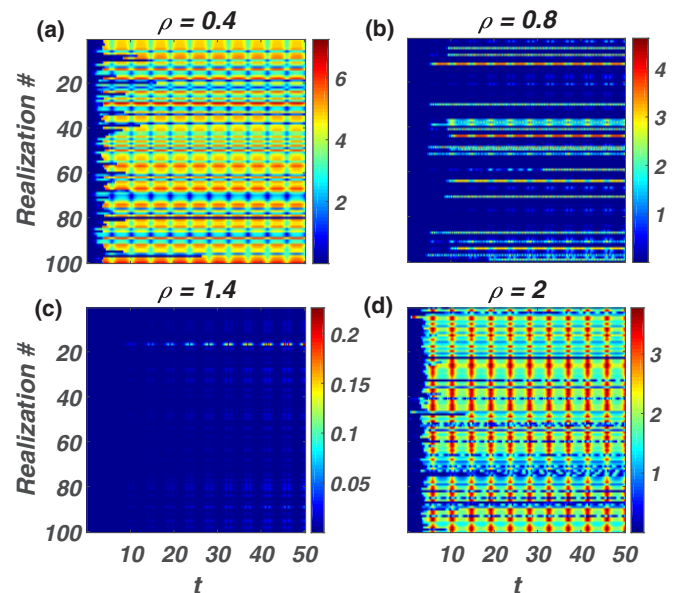


FIG. 11. Time evolution of the RMSE for different statistical realizations in predicting Akhmediev breathers in the NLSE. Four cases are shown, each for a fixed ρ value. For ρ inside the valley interval, most or all realizations exhibit small errors, as in (b) and (c). For ρ outside the interval, almost all realizations exhibit large errors, as in (a) and (d).

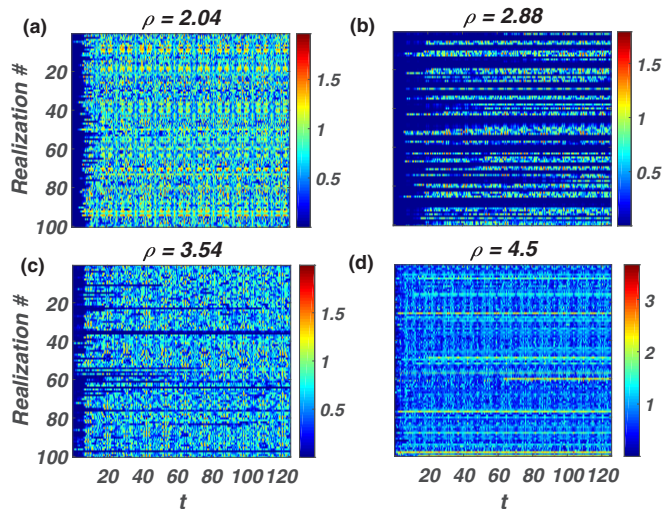


FIG. 12. Time evolution of the RMSE for different statistical realizations in predicting soliton collisions in the NLSE. Parameters of the NLSE solution are $a_1 = 0.14$, $a_2 = 0.34$. Legends are the same as in Fig. 11.

cases outside the valley interval [Figs. 11(a) and 11(d)], the prediction error is large across almost all the realizations. For Fig. 11(c), $\rho = 1.4$, the error is small for almost all the realizations, corresponding to the optimal ρ value in the valley. When ρ deviates from the optimal value, large errors arise with some realizations, as shown in Fig. 11(b) for $\rho = 0.8$. When the majority of the realizations exhibit large errors, the corresponding ρ value is regarded as being outside the valley interval. The error variations across different realizations are characterized by the standard deviation in the mean error. For the optimal ρ value, the standard deviation reaches a

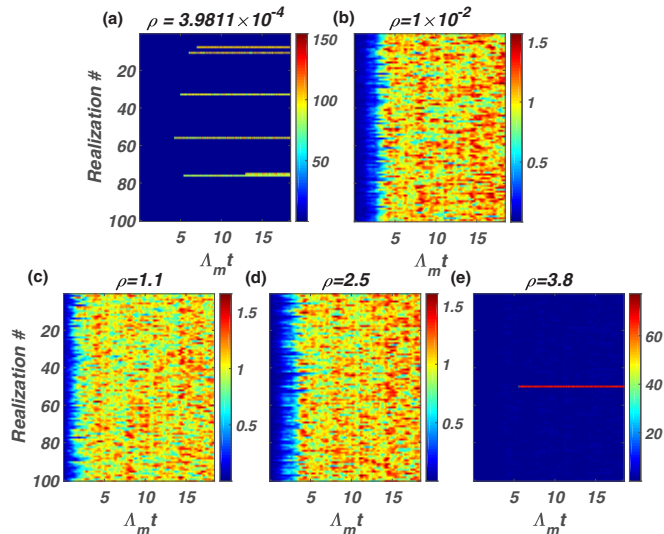


FIG. 13. Time evolution of the RMSE for different statistical realizations in predicting the spatiotemporal chaotic state of the 1D CGLE. Parameter values of the CGLE are the same as for Fig. 5. Legends are the same as in Fig. 11.

minimum. For ρ away from the optimal value, the standard deviation tends to increase. We also note that the concept of the valley interval is meaningful only in an approximate sense: neither the ensemble-averaged error nor the associated standard deviation presents any abrupt changes that can be used to define sharp boundaries of the valley interval.

The variations of the prediction error across individual realizations for the soliton-collision state in the NLSE are shown in Fig. 12, and the corresponding behaviors for predicting the spatiotemporal chaotic state of the 1D CGLE are shown in Fig. 13.

APPENDIX D: SOLUTION OF SOLITON COLLISION IN THE NLSE

The complete solution of soliton collision in the NLSE is given by [20,21]

$$\psi_{12}(x, t) = \psi_0 + \frac{2(l_1^* - l_1)s_1r_1^*}{|r_1|^2 + |s_1|^2} + \frac{2(l_2^* - l_2)s_{12}r_{12}^*}{|r_{12}|^2 + |s_{12}|^2}, \tag{D1}$$

$$r_1(x, t) = \exp\left(\frac{-ix}{2}\right) \left\{ \exp\left[\frac{i(2\chi_1 + \kappa_1 t - \pi/2 + l_1\kappa_1 x)}{2}\right] - \exp\left[\frac{i(-2\chi_1 - \kappa_1 t + \pi/2 - l_1\kappa_1 x)}{2}\right] \right\}, \tag{D2}$$

$$s_1(x, t) = \exp\left(\frac{ix}{2}\right) \left\{ \exp\left[\frac{i(-2\chi_1 + \kappa_1 t - \pi/2 + l_1\kappa_1 x)}{2}\right] + \exp\left[\frac{i(2\chi_1 - \kappa_1 t + \pi/2 - l_1\kappa_1 x)}{2}\right] \right\}, \tag{D3}$$

$$r_2(x, t) = \exp\left(\frac{-ix}{2}\right) \left\{ \exp\left[\frac{i(2\chi_2 + \kappa_2 t - \pi/2 + l_2\kappa_2 x)}{2}\right] - \exp\left[\frac{i(-2\chi_2 - \kappa_2 t + \pi/2 - l_2\kappa_2 x)}{2}\right] \right\}, \tag{D4}$$

$$s_2(x, t) = \exp\left(\frac{ix}{2}\right) \left\{ \exp\left[\frac{i(-2\chi_2 + \kappa_2 t - \pi/2 + l_2\kappa_2 x)}{2}\right] + \exp\left[\frac{i(2\chi_2 - \kappa_2 t + \pi/2 - l_2\kappa_2 x)}{2}\right] \right\}, \tag{D5}$$

$$r_{12}(x, t) = \frac{(l_1^* - l_1)s_1^*r_1s_2 + (l_2 - l_1)|r_1|^2r_2 + (l_2 - l_1^*)|s_1|^2r_2}{|r_1|^2 + |s_1|^2}, \tag{D6}$$

$$s_{12}(x, t) = \frac{(l_1^* - l_1)s_1r_1^*r_2 + (l_2 - l_1)|s_1|^2s_2 + (l_2 - l_1^*)|r_1|^2s_2}{|r_1|^2 + |s_1|^2}, \tag{D7}$$

$$\psi_0(x, t) = \exp(ix), \tag{D8}$$

where $l_1 = i\sqrt{2a_1}$, $l_2 = i\sqrt{2a_2}$, $\kappa_1 = 2\sqrt{1+l_1^2}$, $\kappa_2 = 2\sqrt{1+l_2^2}$, $\chi_1 = \frac{1}{2} \arccos(\kappa_1/2)$, $\chi_2 = \frac{1}{2} \arccos(\kappa_2/2)$, and $*$ represents the complex conjugate.

-
- [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] 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).
- [6] 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).
- [7] T. L. Carroll, Using reservoir computers to distinguish chaotic signals, *Phys. Rev. E* **98**, 052209 (2018).
- [8] K. Nakai and Y. Saiki, Machine-learning inference of fluid variables from data using reservoir computing, *Phys. Rev. E* **98**, 023111 (2018).
- [9] Z. S. Roland and U. Parlitz, Observing spatio-temporal dynamics of excitable media using reservoir computing, *Chaos* **28**, 043118 (2018).
- [10] 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).
- [11] H. Jaeger, The “echo state” approach to analyzing and training recurrent neural networks—with an erratum note, GMD Technical Report No. 148, German National Research Center for Information Technology, Bonn (2001), p. 13.
- [12] G. Manjunath and H. Jaeger, Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks, *Neur. Comp.* **25**, 671 (2013).
- [13] W. Mass, T. Nachtschlaeger, and H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, *Neur. Comp.* **14**, 2531 (2002).
- [14] H. Jaeger and H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, *Science* **304**, 78 (2004).
- [15] J. M. Dudley, F. Dias, M. Erkintalo, and G. Genty, Instabilities, breathers and rogue waves in optics, *Nat. Photon.* **8**, 755 (2014).
- [16] D. R. Solli, C. Ropers, P. Koonath, and B. Jalali, Optical rogue waves, *Nature* **450**, 1054 (2007).
- [17] J. M. Dudley, G. Genty, F. Dias, B. Kibler, and N. Akhmediev, Modulation instability, Akhmediev breathers and continuous wave supercontinuum generation, *Opt. Express* **17**, 21497 (2009).
- [18] B. Kibler, J. Fatome, C. Finot, G. Millot, F. Dias, G. Genty, N. Akhmediev, and J. M. Dudley, The Peregrine soliton in nonlinear fibre optics, *Nat. Phys.* **6**, 790 (2010).
- [19] N. Akhmediev and V. Korneev, Modulation instability and periodic solutions of the nonlinear Schrödinger equation, *Theor. Math. Phys.* **69**, 1089 (1986).
- [20] N. Akhmediev, J. M. Soto-Crespo, and A. Ankiewicz, Extreme waves that appear from nowhere: On the nature of rogue waves, *Phys. Lett. A* **373**, 2137 (2009).
- [21] B. Frisquet, B. Kibler, and G. Millot, Collision of Akhmediev Breathers in Nonlinear Fiber Optics, *Phys. Rev. X* **3**, 041032 (2013).
- [22] I. E. Papacharalampous, P. G. Kevrekidis, B. A. Malomed, and D. J. Frantzeskakis, Soliton collisions in the discrete nonlinear Schrödinger equation, *Phys. Rev. E* **68**, 046604 (2003).
- [23] I. S. Aranson and L. Kramer, The world of the complex Ginzburg-Landau equation, *Rev. Mod. Phys.* **74**, 99 (2002).
- [24] M. C. Cross and P. C. Hohenberg, Pattern formation outside of equilibrium, *Rev. Mod. Phys.* **65**, 851 (1993).
- [25] Y. Kuramoto, *Chemical Oscillations, Waves and Turbulence* (Springer, Berlin, 1984).
- [26] J.-W. Kim and E. Ott, Statistics and characteristics of spatiotemporally rare intense events in complex Ginzburg-Landau models, *Phys. Rev. E* **67**, 026203 (2003).
- [27] V. Nagy and E. Ott, Control of rare intense events in spatiotemporally chaotic systems, *Phys. Rev. E* **76**, 066206 (2007).
- [28] L. Du, Q. Chen, Y.-C. Lai, and W. Xu, Observation-based control of rare intense events in the complex Ginzburg-Landau equation, *Phys. Rev. E* **78**, 015201(R) (2008).
- [29] S. M. Cox and P. C. Matthews, Exponential time differencing for stiff systems, *J. Comput. Phys.* **176**, 430 (2002).
- [30] J. D. Farmer and J. J. Sidorowich, Predicting Chaotic Time Series, *Phys. Rev. Lett.* **59**, 845 (1987).
- [31] M. Casdagli, Nonlinear prediction of chaotic time series, *Physica D* **35**, 335 (1989).
- [32] J. B. Elsner and A. A. Tsonis, Nonlinear prediction, chaos, and noise, *Bull. Am. Meteorol. Soc.* **73**, 49 (1992).
- [33] V. Petrov, Nonlinear prediction, filtering, and control of chemical systems from time series, *Chaos* **7**, 614 (1997).
- [34] F. Takens, Detecting strange attractors in fluid turbulence, in *Dynamical Systems and Turbulence. Lecture Notes in Mathematics*, Vol. 898, edited by D. Rand and L. S. Young (Springer-Verlag, Berlin, 1981), pp. 366–381.

- [35] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovavis, and C. Grebogi, Predicting Catastrophes in Nonlinear Dynamical Systems by Compressive Sensing, *Phys. Rev. Lett.* **106**, 154101 (2011).
- [36] 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).
- [37] M. Lukoševičius and H. Jaeger, Reservoir computing approaches to recurrent neural network training, *Comput. Sci. Rev.* **3**, 127 (2009).
- [38] M. Lukoševičius, A practical guide to applying echo state networks, in *Neural Networks: Tricks of the Trade* (Springer, Berlin, 2012), pp. 659–686.