Machine-learning parameter tracking with partial state observation

Zheng-Meng Zhai,[1] Mohammadamin Moradi,[1] Bryan Glaz,[2] Mulugeta Haile,[3] and Ying-Cheng Lai[1,4,*]

[1]*School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, Arizona 85287, USA*
[2]*Army Research Directorate, DEVCOM Army Research Laboratory, Adelphi, Maryland 20783-1138, USA*
[3]*Army Research Directorate, DEVCOM Army Research Laboratory, Aberdeen Proving Ground, Maryland 21005-5069, USA*
[4]*Department of Physics, Arizona State University, Tempe, Arizona 85287, USA*

(Received 24 October 2023; accepted 28 January 2024; published 23 February 2024)

Complex and nonlinear dynamical systems often involve parameters that change with time, accurate tracking of which is essential to tasks such as state estimation, prediction, and control. Existing machine-learning methods require full state observation of the underlying system and tacitly assume adiabatic changes in the parameter. Formulating an inverse problem and exploiting reservoir computing, we develop a model-free and fully data-driven framework to accurately track time-varying parameters from partial state observation in real time. In particular, with training data from a subset of the dynamical variables of the system for a small number of known parameter values, the framework is able to accurately predict the parameter variations in time. Low- and high-dimensional, Markovian and non-Markovian, and spatiotemporal nonlinear dynamical systems are used to demonstrate the power of the machine-learning based parameter-tracking framework. Pertinent issues affecting the tracking performance are addressed.

DOI: 10.1103/PhysRevResearch.6.013196

I. INTRODUCTION

The behavior of a nonlinear dynamical system is controlled by its parameters. In a real-world environment, the parameters typically change or drift with time. For example, when an optical sensor system is deployed to an outdoor environment, climatic disturbances such as temperature and humidity fluctuations can cause the geometrical and material parameters of the system to change with time. In an ecological system, seasonal fluctuations and human influences on the environment can induce changes in the parameters underlying the population dynamics such as the carrying capacity and species decay rates. Often, due to the complex interactions between the system and the environment, the simplistic assumption that the parameters drift linearly with time is not valid. Rather, the variations of the parameters with time can be complicated. A generic feature of nonlinear dynamical systems is that even a small parameter change can lead to characteristically different and even catastrophic behaviors. For example, a nonlinear system can typically exhibit a variety of bifurcations including a crisis [1] at which a chaotic attractor is destroyed and replaced by transient chaos [2], leading to system collapse. Being able to predict or forecast how some key system parameters change with time into the future can lead to control strategies to prevent system collapse.

The problem of tracking parameter variations is an inverse problem [3], which is difficult even if an accurate mathematical model of the system is known. A related concept is data assimilation [4,5] that aims to tackle challenges in estimating the state and certain parameters. Our assumption is that the parameter of interest cannot be directly accessed or measured, so tracking its variations will need to be done indirectly using the measurements of some accessible dynamical variables of the system. Suppose that a key parameter will change with time in the future but at present the system is stationary so that a few distinct values of this parameter can be measured, together with the time series of a subset of the dynamical variables. A scenario is that an instrument or device is to be deployed in certain missions where the harsh and nonstationary environment will cause the key parameter to change with time. Before deployment, the device can be tested in a controlled laboratory environment where the values of the parameter and the corresponding time series can be obtained. Assuming in the real environment the parameter cannot be measured but some time series from partial state observation still can be, we ask the question of whether it is possible to extract the parameter variations. In this work, we demonstrate that machine learning can be exploited to provide an affirmative answer.

The idea of exploiting machine learning for parameter extraction has been investigated recently [6–11] (Sec. II provides a background review of the previous methods on static parameter identification and dynamic parameter tracking, as well as the more recent machine-learning approaches). In existing machine-learning works, the time series from all dynamical variables and the time variations of the parameter is required for training. Here we articulate a machine-learning framework with the following two main features that go beyond the

*Ying-Cheng.Lai@asu.edu

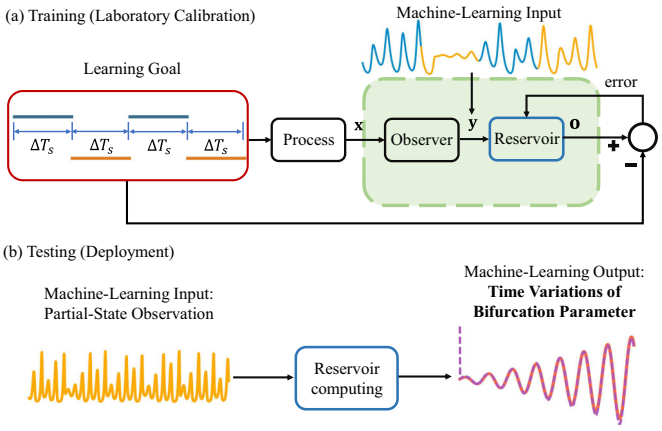

FIG. 1. Proposed reservoir-computing based parameter tracking scheme. The goal is to track a parameter of the system from which only partial state observation is available. The output of the reservoir computer is $o(t)$. (a) In the training or laboratory calibration phase, the input to the neural network is an integrated vector time series $\mathbf{y}_T(t)$ recombining segments of the time series from partial state observation for a small number of parameter values. The training goal is the corresponding piecewise constant function of the parameter values versus time, denoted as $p(t)$, i.e., minimizing the error between $p(t)$ and $o(t)$. (b) In the testing or deployment phase where the parameters are inaccessible, the input to the neural network is the current observation vector time series $\mathbf{y}(t)$ and the output is the parameter as a function of time.

existing methods: (1) only the measurements from a partial set of the dynamical variables are needed, (2) observation of the state from a small number of parameter values suffices, and (3) the historical parameter values are not required in real-time parameter tracking. More specifically, let $\mathbf{x} \in \mathcal{R}^D$ be the D-dimensional state vector of the dynamical system and let $\mathbf{y} \in \mathcal{R}^{D'}$ be the measurement or observation vector: $\mathbf{y} = \mathbf{g}(\mathbf{x})$, where $D' < D$ and $\mathbf{g} : \mathcal{R}^D \to \mathcal{R}^{D'}$ is the measurement function. We choose reservoir computing [12–15] as the machine-learning architecture, which in recent years has been applied to predicting nonlinear dynamical systems [16–40], and propose the following parameter tracking scheme. Suppose the goal is to track a single parameter p (for simplicity), so the output of the neural network is a scalar quantity $o(t)$. In a well-controlled laboratory environment, vector time series of dimension D' from a small number of parameter values can be measured. We construct the input data by first breaking the measured time series into a number of segments of equal length and recombining them to form an integrated vector time series $\mathbf{y}(t)$. Corresponding to each segment in $\mathbf{y}(t)$, there is an exact value of the parameter, thereby generating a piecewise constant function of the parameter with time, denoted as $p(t)$. The goal of training is to minimize the error between $o(t)$ and $p(t)$, as shown in Fig. 1(a). This arrangement ensures that the neural network learns the dynamical "climate" of the underlying system and how it changes with time through alternating exposure to the measurements taking from different parameter values. During the testing phase, e.g., when the system is deployed to a real application environment, the parameter varies with time and it is no longer accessible to observation or measurement. What can be observed is vector time series $\mathbf{y}(t)$. When the well-trained neural network takes in $\mathbf{y}(t)$ as the input, its output should give the time variation of the parameter, realizing accurate parameter tracking, as illustrated in Fig. 1(b).

In Sec. II, we present a concise background review of the main existing approaches to parameter identification and tracking. Success of parameter tracking is demonstrated in Sec. III using prototypical, low- and high-dimensional, Markovian and non-Markovian nonlinear dynamical systems: a low-dimensional chaotic food-chain system, the chaotic Rössler oscillator, the high-dimensional chaotic Mackay-Glass delay-differential equation system, and the spatiotemporal Lorenz-96 model. Conclusions and discussion are offered in Sec. IV.

II. BACKGROUND LITERATURE REVIEW

There are two types of tasks associated with the general problem of parameter extraction: identification and tracking. For parameter identification, the parameter of interest is a constant, and the task is to estimate its value. The problem of parameter tracking is more difficult, where the parameter is time varying and the task is to predict the parameter as a function of time.

Most existing works dealt with the parameter-identification problem, and a wide variety of methods were proposed in different fields. The conventional least-squares method [41] for parameter identification fits a linear or a nonlinear model to data by minimizing the sum of the squares of the errors. Likewise, the method of maximum likelihood estimation [42] finds the set of parameter values which, given the model, maximizes the likelihood of the data. The maximum-likelihood method is often used in statistical modeling and can be applied to both continuous-time and discrete-time systems. The method of Bayesian estimation [43] uses the Bayes's theorem to update the probability of a set of parameter values based on new data, which is commonly used in statistical modeling and enables incorporating prior knowledge about the parameters into the model. Genetic algorithms [44] exploit the principles of natural selection and genetics to optimize a continuous-time or discrete-time model to identify the parameter values. For identifying parameters associated with complex patterns from large datasets, artificial neural networks can be used [45]. The method of Markov-chain Monte Carlo [46] involves sampling from a distribution of the possible parameter values to estimate the posterior distribution, which is commonly used in statistical modeling and allows for the incorporation of uncertainty in the parameter estimates. A widely used parameter-estimation method is Kalman filters [47–51]. It deals with linear or nonlinear dynamical systems with an uncertain state-space model through a combination of prediction and correction steps. Specifically, at the prediction step, the Kalman filter uses the current estimate of the parameter values and the dynamics to predict the future state of the system. At the correction step, the Kalman filter compares the predicted state with the measured state and uses the difference between them to update the estimate of the parameters. This process is repeated iteratively, and the estimates of the parameters are updated at each iteration based on the measurement residuals.

When the parameter of interest is hidden in the sense that it cannot be directly measured or accessed in an specific application, machine learning provides a viable solution [52]. Specifically, given time series of the dynamical variables of the system, a machine-learning algorithm can identify the time averaged value of the parameter.

Computationally, parameter identification can be done through a single scan [53] or multiple scans [54]. In the single-scan scheme, the parameters are estimated using a single measurement or a single set of measurements [53]. This approach was often used in situations where it is not possible or practical to take multiple measurements or where the system changes so rapidly to render multiple measurements not representative. The methodology can be either static or dynamic, depending on whether the system being measured is time invariant or time varying. In particular, static single-scan estimation can be done using least-squares fitting or numerical optimization techniques such as gradient descent. Multiscan parameter estimation [54] is based on taking multiple measurements of the system over a period of time. It can be more accurate than single-scan estimation through some averaging process of the measurements and can detect parameter variations with time. However, the method is resource and computation intensive, due to the need to take multiple measurements and the required computation.

The problem of tracking the time variations of a parameter is significantly more challenging than static parameter identification. Earlier, machine learning was used for integration-free and data-driven extraction of a set of spatiotemporal stride parameters [6]. Recently, a machine-learning method was proposed [7] for this task based on measured time-series data. In the training phase, time series from all dynamical variables of the system are required and, in the testing phase, the exact values of the parameters in the past are required. A common difficulty in using neural networks to extract parameter variations is the slow convergence rate and local optima stagnation in solving the underlying complex optimization problem. In another recent work, an improved machine-learning model (neural network algorithm with reinforcement learning) was proposed for extracting parameters of photovoltaic models [9]. More recently, a machine-learning based approach to feature parameter extraction from speech signals to improve the performance of speech recognition applications in real-time smart city environments was articulated [10]. In general, difficulties associated with certain supervised machine-learning methods include large sample demand, high computational complexity, and low accuracy when processing high-dimensional data. One solution is the manifold Gaussian process machine-learning method based on the differential evolution algorithm [8] to extract the dimension-reduction parameters. In addition, a deep learning-based parameter extraction method was proposed [11].

III. RESULTS

We consider four prototypical nonlinear dynamical systems: a three-species chaotic food-chain system [55], the chaotic Rössler oscillator [56], the Mackey-Glass delay-differential equation system [57], and the 40-dimensional Lorenz-96 system [58]. The first two systems are three-dimensional while the Mackey-Glass system is non-Markovian with an infinite-dimensional phase space. The Lorenz-96 system is a high-dimensional spatiotemporal system. For each system, three types of parameter variations are considered: frequency modulation (FM), sawtooth wave, and amplitude modulation (AM), with different numbers of parameter sampling in different ranges for training. In all cases, partial state observation is used. The machine-learning performance is characterized by the root-mean-square error (RMSE). The effects of measurement and dynamical noises as well as the hyperparameters on the parameter-tracking performance are also studied. Simulations are run using MATLAB on a desktop computer with 32 CPU cores, 128-GB memory, and one RTX 4000 NVIDIA GPU.

For clarity of presentation, we show the results for the chaotic food-chain system here while displaying those from the chaotic Rössler, Mackey-Glass and Lorenz-96 systems in Appendices B, C, and D, respectively.

A. Visualization of parameter-tracking results

Our first example is a food-chain system of three species: resource, consumer, and predator [55], described by the following set of nonlinear differential equations:

$$\begin{aligned}\frac{dR}{dt} &= R\left(1-\frac{R}{K}\right)-\frac{x_c y_c C R}{R+R_0},\\ \frac{dC}{dt} &= x_c C\left(\frac{y_c R}{R+R_0}-1\right)-\frac{x_p y_p P C}{C+C_0},\\ \frac{dP}{dt} &= x_p P\left(\frac{y_p C}{C+C_0}-1\right),\end{aligned} \tag{1}$$

where R, C, and P are the population densities of the resource, consumer, and predator species, respectively. The system has seven parameters: $K, x_c, y_c, x_p, y_p, R_0, C_0 > 0$. To illustrate the process of parameter tracking, we assume that the following three parameters: K, y_c, and y_p are time varying, and fix all other parameters at constant values: $x_c = 0.4$, $x_p = 0.08$, $C_0 = 0.5$, and $R_0 = 0.16129$, according to some bioenergetics argument [55]. We focus on tracking a single parameter: K, y_c, or y_p, whose respective nominal values are 0.94, 1.7, and 5.0. When tracking one of the three parameters, the other two are fixed at their respective nominal values. Representative bifurcation diagrams of the food-chain system with respect to the three parameters: K, y_c, y_p are shown in Fig. 2.

Figure 3 exemplifies the results of tracking the three different types of parameter variations for different combinations of the bifurcation parameter and state observation. The computational setting is as follows. Time series are generated by integrating the system model using the time step $dt = 0.01$. The initial states of both the dynamical process and the reservoir neural network are randomly chosen from a uniform distribution. The training and testing data are obtained by sampling the time series at the interval Δ_s. We set $\Delta_s = 250dt = 2.5$, corresponding to approximately 1/25 cycles of oscillation in the chaotic food-chain system. Let ΔT_s be the switching time interval in which the target parameter is a constant. The training time is $900\Delta T_s$. The testing length is chosen to be slightly longer than the training length, enough for tracking several cycles of the parameter variation.

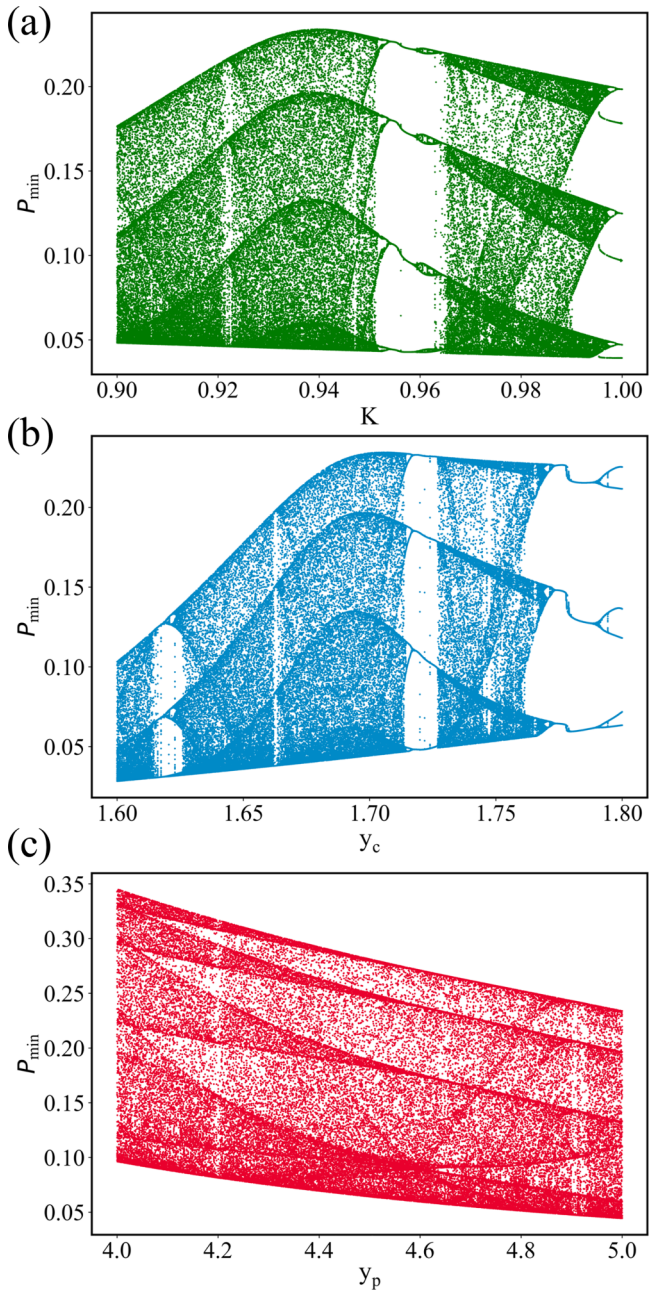

FIG. 2. Bifurcation diagrams of the chaotic food chain system. [(a)–(c)] Diagrams for the bifurcation parameters K, y_c, and y_p, respectively.

The size of the reservoir network is $D_r = 500$ and the bias number is set to 1. The other six hyperparameters are determined by Bayesian optimization (Appendix A), which are fixed during training and testing. The values of the optimized hyperparameters for the chaotic food-chain system are listed in Table I. Often, a constant bias may arise between the machine predicted parameter variation and the ground truth, which can be removed by calibrating using two parameter values (Appendix E). The results in Fig. 3 indicate that the machine-learning framework is capable of accurate parameter tracking based on partial state observation only.

TABLE I. Optimal values of the reservoir-computing hyperparameters for tracking the variations of three different bifurcation parameters of the chaotic food-chain system (1).

Bifurcation parameter	ρ	γ	α	β	p
K	1.47	0.04	0.93	$10^{-5.8}$	0.67
y_c	0.42	0.40	0.21	$10^{-3.7}$	0.21
y_p	1.92	0.10	0.65	$10^{-3.9}$	0.70

B. Minimally required bifurcation parameter values for training

Two characteristics of the bifurcation parameter values used in the training which can affect significantly the performance of tracking are the number of such parameter values (denoted as s_n) and the relative range of the parameter variation (denoted as s_w) from which the training time-series data are generated. Intuitively, if the training data come from only one value of the bifurcation parameter, then it will not be possible for the reservoir computer to learn the features of parameter variation, resulting in a large tracking error. As s_n increases from 1, we expect the error to decrease. What is the minimum number of bifurcation parameter values required for accurate parameter tracking? Likewise, if the bifurcation parameter values are taken from the full range of the actual parameter variation, then accurate tracking is likely, resulting in a small error. As the range s_w is reduced, the error will increase. How tolerant can the machine-learning parameter tracking scheme be with respect to this range?

Figures 4(a), 4(c), and 4(e) demonstrate the effect of varying s_n on the parameter-tracking performance for different combinations of the bifurcation parameter and its time variations. It can be seen that, for all cases illustrated, the testing error decreases dramatically as s_n increases from one to three and remains approximately constant afterwards, indicating that using the time series from as few as three values of the bifurcation parameter suffices for accurate tracking of the actual parameter variations. Figures 4(b), 4(d), and 4(f) show the effect of varying the relative range s_w on the tracking performance. As s_w decreases from 100%, the RMSE increases, but it does so in a slow manner until s_w falls below about 20%, indicating that the machine-learning parameter tracking scheme is remarkably tolerant to the range of the bifurcation parameter from which the time series for training are generated.

C. How partial can state observation be?

The main feature of our work is that tracking complicated time variations of a bifurcation parameter can be achieved using only partial state observation by exploiting machine learning. Historically, the celebrated and extensively applied Takens's delay-coordinate embedding theorem [59] provided the mathematical foundation to reconstruct the phase space of a dynamical system even with a single measured time series. Indeed, Takens' methodology guarantees faithful reconstruction of a topological equivalent of the underlying dynamical system, which allows the key dynamical invariants such as the Lyapunov exponents and the fractal dimensions of the attractor to be estimated. Our task of accurate parameter

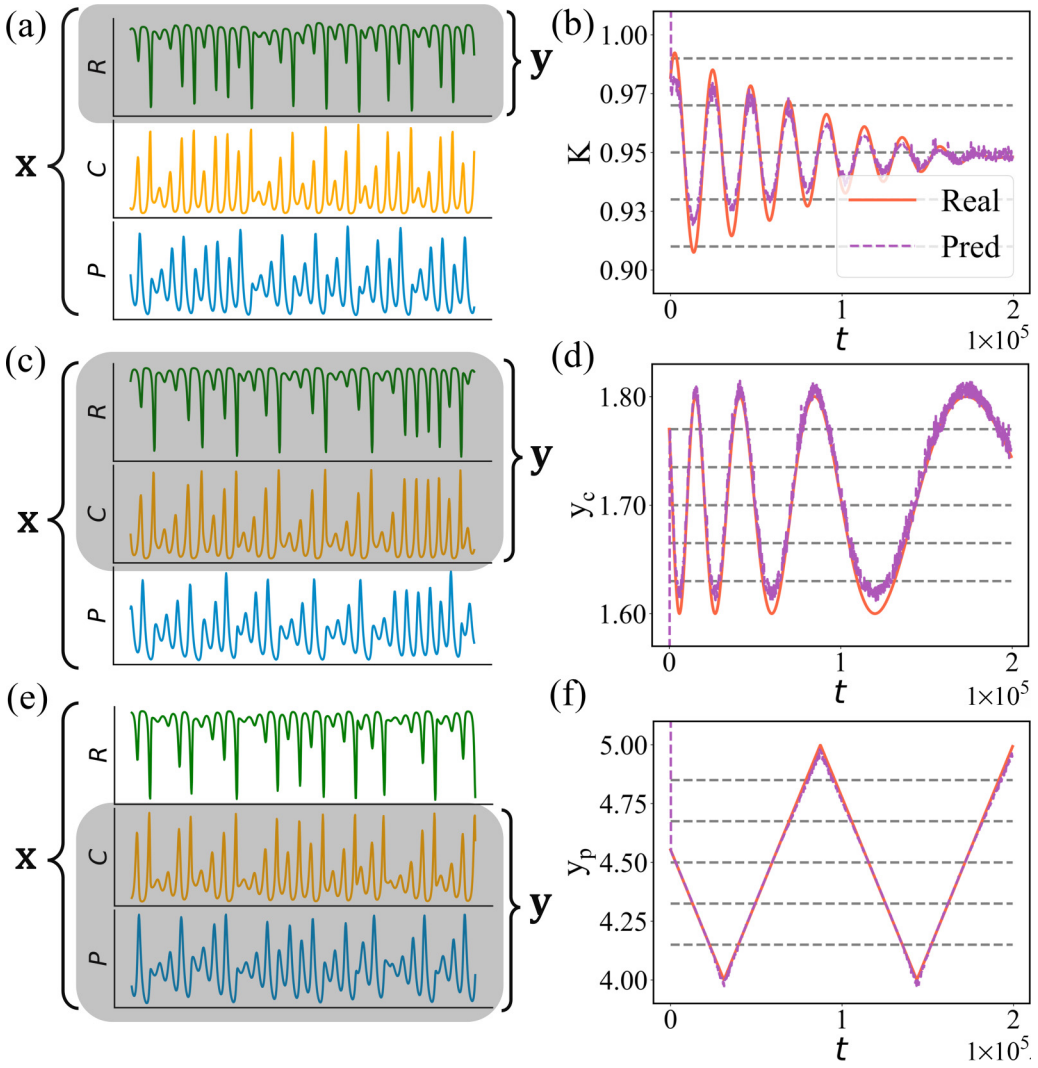

FIG. 3. Tracking time-varying parameters of the chaotic food-chain system. Different combinations of the parameter waveforms and partial state observation are illustrated: the top, middle, and bottom row correspond to three types of parameter variations (AM, FM, and sawtooth waveform), while the gray-shaded region in the left column illustrates the partial state observation. The right column gives the results of parameter tracking in comparison with the ground truth. [(a) and (b)] One-dimensional observational state variable $\mathbf{y} = (R)$ for tracking AM parameter K. [(c) and (d)] Two-dimensional observation $\mathbf{y} = (R, C)$ for tracking FM parameter y_c. [(e) and (f)] Two-dimensional observation $\mathbf{y} = (C, P)$ for tracking the sawtooth-wave parameter y_p. The five horizontal dashed lines indicate the parameter values from which training data are generated, i.e., $s_n = 5$.

tracking, where the bifurcation parameter is assumed to vary with time in a sophisticated manner, requires reconstruction of the system at a more detailed level, for which the embedding methodology is not effective. A key question is how partial the state observation can be. A related question is which "typical" variables can be observed to ensure accurate parameter tracking. Atypical situations can often arise; for example, the z variable of the Lorenz-63 chaotic attractor cannot be used to infer the x and y variables, due to an inherent symmetry of the system. In nonlinear dynamics, a general theory for determining a set of partial observations for obtaining the complete knowledge of the whole system has not been available, yet. We thus resort to a systematic numerical approach to addressing these questions by exploring a range of variables and testing their efficacy in parameter tracking. In particular, for $\mathbf{x} \equiv (x_1, x_2, x_3)^T$, full state observation is represented by $\mathbf{y} = \mathbf{x}$. There are six cases of partial state observation: $\mathbf{y} = (x_1)$, $\mathbf{y} = (x_2)$, $\mathbf{y} = (x_3)$, $\mathbf{y} = (x_1, x_2)$, $\mathbf{y} = (x_1, x_3)$, and $\mathbf{y} = (x_2, x_3)$. Altogether, there are seven distinct cases of state observation. For each case, we conduct a reasonable of number (e.g., 50) of tests. For each test, define an error threshold. If the resulting testing RMSE is below the threshold, then the test is deemed successful. The fraction of successful cases gives the success probability. We then calculate, for different types of parameter variations, the success rate P_s and use a bar graph to represent the rate for the seven cases of state observation. Some representative results for the chaotic food-chain system are shown in Fig. 5. It can be seen that in most cases, close to 100% success rate can be achieved by observing two state variables. Depending on the specific parameter, in some

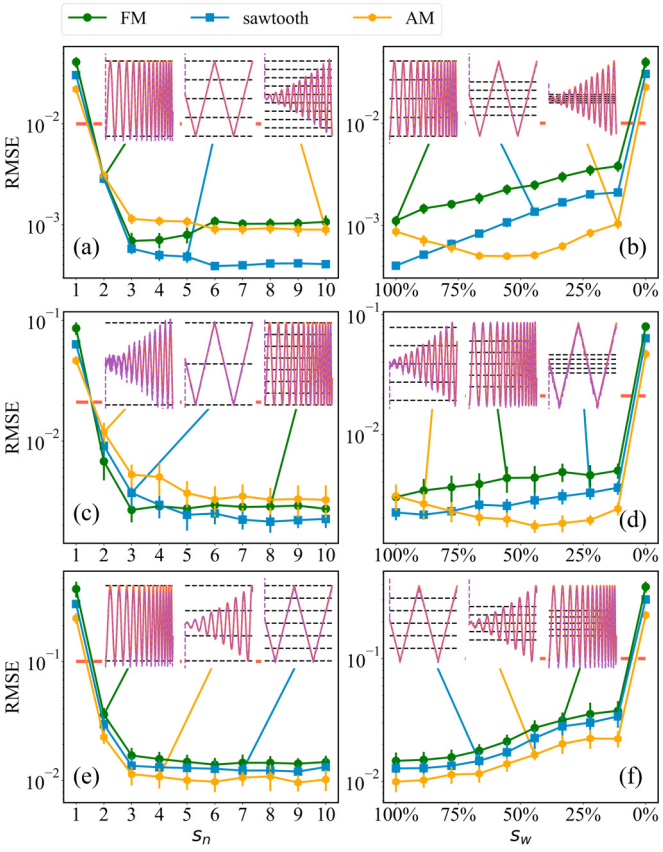

FIG. 4. Effects of the training bifurcation-parameter values on tracking. The two relevant quantities are s_n (the number of distinct bifurcation-parameter values for training) and s_w (the relative parameter range from which time series are taken for training). [(a), (c), and (e)] Testing RMSE versus s_n for parameters K, y_c, and y_p, respectively. [(b), (d), and (f)] Testing RMSE versus s_w for K, y_c, and y_p, respectively. Each panel has results from FM, sawtooth, and AM types of parameter variations, and each point is the result of averaging over 50 independent realizations. The results in (a), (c), and (e) indicates that using time series from three distinct values of the bifurcation parameter suffices to guarantee accurate parameter tracking. The results in (b), (d), and (f) suggests acceptable parameter-tracking performance for $s_w > 20\%$.

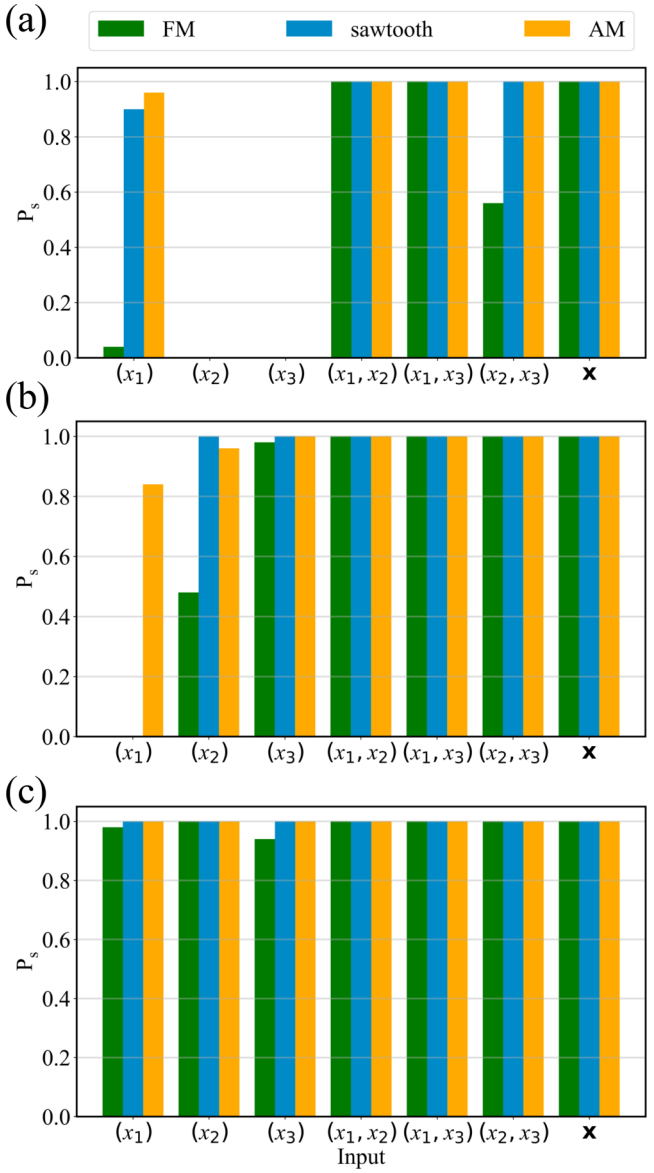

FIG. 5. Success probability for different partial observation scenarios for the chaotic food-chain system. [(a)–(c)] Bar-graph representation of the success rate for the seven distinct cases of state observation for the three different bifurcation parameters, for three types of parameter variation (FM: green; sawtooth: blue; AM: yellow). The success rate is calculated from 50 independent realizations. In most cases, observing two state variables can lead to nearly 100% success rate.

cases observing even one state variable can lead to satisfactory success rate.

In nonlinear dynamical systems, the situation of "atypical" variables can often arise in the sense that the observation of such a variable cannot be used to construct the original system dynamics. Examples are the z variable in the classical Lorenz-63 system or the Rössler oscillator, which cannot be used to infer the dynamical behaviors of the variables x and y. One reason for reconstruction to fail is an inherent symmetry of the system, which is difficult to detect, especially in realistic situations where the detailed system equations are not known. At the present, a theory to determine whether a specific configuration of partial observation is enough to reconstruct the dynamics or track the parameters in nonlinear dynamical systems is not available. Our approach is thus computational: As a proof of principle, we apply the machine-learning based framework to some classical chaotic systems to demonstrate that accurate tracking of a parameter varying in time in a complicated way is indeed possible, but failure can arise if the observed variable is "atypical" in the aforementioned sense. For instance, for the chaotic food-chain system, the success rate of tracking the time variations of the bifurcation parameter K by observing $\mathbf{y} = (x_2)$ or $\mathbf{y} = (x_3)$ is zero, as shown in Fig. 5.

D. Robustness against noise

It is important to study how parameter-tracking performance is degraded by measurement and dynamical noises.

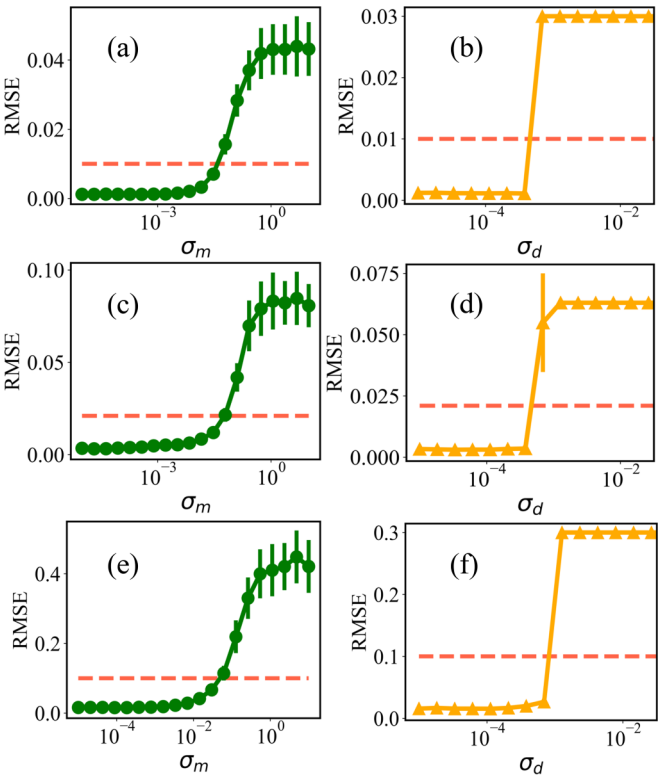

FIG. 6. Robustness of parameter tracking against measurement and dynamical noises. The upper, middle, and lower rows correspond to tracking the three parameters (K, y_c, and y_p) in the chaotic food-chain system, respectively. The left and right columns are for measurement and dynamical noises, respectively. FM type of parameter variations is used. Each value of the testing RMSE and the error bar are obtained from 50 independent training and test runs. The RMSEs remain small and approximately constant if the noise amplitude is below some reasonable value. The horizontal dashed lines are indicative of some (arbitrary) threshold below which the tracking performance is deemed satisfactory.

Measurement noise can be modeled as additive noise that perturbs the data vector $\mathbf{x}$ to $\mathbf{x} + \xi_n$ before normalization, and dynamical noise can be incorporated into the system dynamics, which perturbs the system function (or velocity field) at each time step [60]. We assume that both types of noises are normally distributed with zero mean and standard deviation σ_m and σ_d, respectively. Figure 6 illustrates the effects of measurement and dynamical noises on time-varying parameter tracking, for the three bifurcation parameters in the food-chain system subject to FM. It can be seen that, for both measurement and dynamical noises, insofar as the noise amplitude is below some reasonable value, the testing RMSE remains small, demonstrating that our machine-learning parameter tracking scheme is robust against the noises.

E. Effect of network size and training time on tracking performance and minimum switching time required of the training data

Figure 7 illustrates the effect of network size D_r and training time T_{train} on tracking performance, where the color-coded values of the success probability are displayed in the two-dimensional parameter plane (D_r, T_{train}). As the network becomes larger, the success probability can be dramatically improved, as can be intuitively expected. However, the performance decreases as the training time increases, due to overfitting, especially in cases where the network size is not large enough.

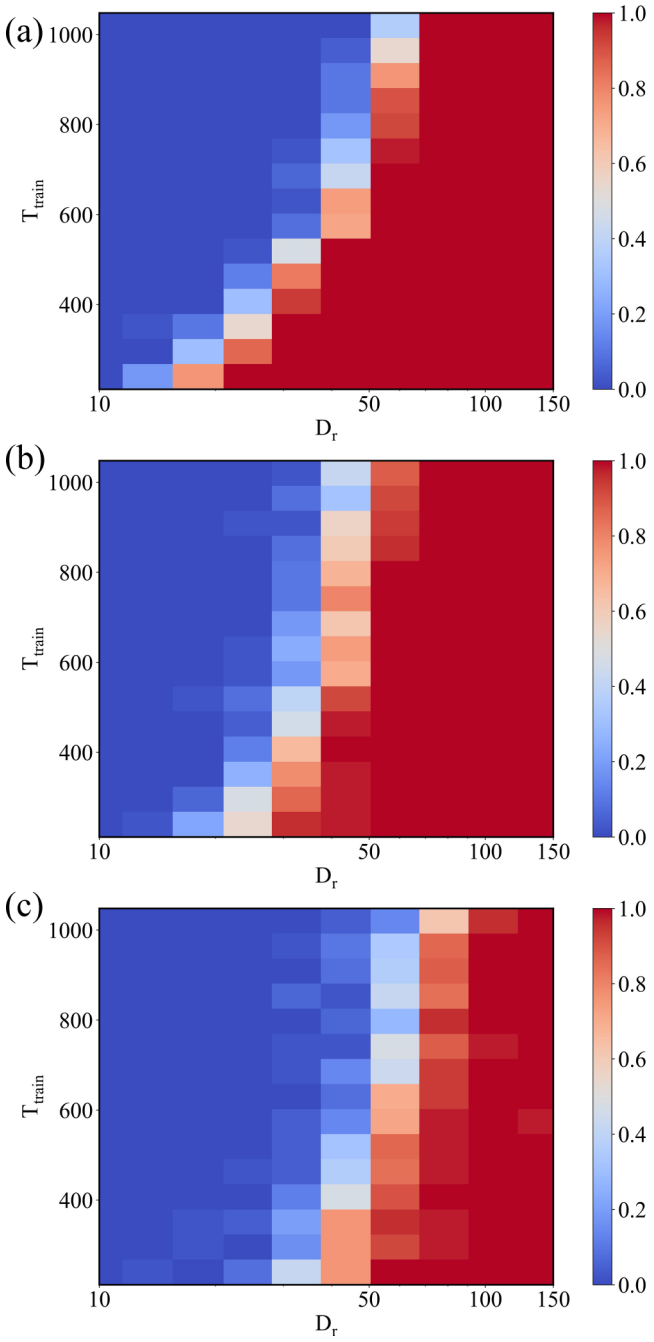

FIG. 7. Effects of two hyperparameters: network size D_r and training time T_{train}, on parameter-tracking performance. [(a)–(c)] Color-coded success probability values in the plane of these two hyperparameters for tracking the bifurcation parameters K, y_c, and y_p in the food chain system, respectively, subject to FM type of variations. Each value of the success probability is obtained using 50 statistical realizations. Increasing the network size and the training time can often dramatically improve the success rate.

In machine learning in general, increasing the training data should reduce overfitting for a model of fixed complexity. However, in reservoir computing, "fixed complexity" is not solely related to the reservoir size but also includes dynamics within the reservoir network. If the network size is not sufficiently large, then the inherent complexity within the reservoir will not be high enough to capture the nuances of the larger dataset, and this can reduce the parameter-tracking performance. In essence, the reservoir computer maps the low-dimensional data into a high-dimensional complex hidden state. The anomaly in Fig. 7 arises mainly due to the fixed complexity in reservoir computing. Empirically, as the network becomes larger, the performance should not worsen insofar as more training data are used.

As indicated in Fig. 1, the training data consist of mutually switched time series from a small number of bifurcation-parameter values, with switching time ΔT_s, the length of each continuous time-series segment. Intuitively, time series of certain length are needed for the neural network to learn the dynamics at the specific bifurcation-parameter value. If ΔT_s is too small, then this would not be possible, even with frequent switchings of the time series from different bifurcation-parameter values. Furthermore, a small ΔT_s indicates a fast change in the bifurcation parameter during the testing phase, rendering difficult for the machine-learning scheme to track the waveform. We thus expect large RMSEs for small ΔT_s values but, as ΔT_s increases, the testing errors will decrease. Figures 8(a)–8(c) show this effect for the three bifurcation parameters (K, y_c, and y_p) of the chaotic food-chain system, respectively, where the switching time ΔT_s is in units of the sampling time interval Δ_s. In each case, three types of parameter variations (FM, sawtooth, AM) are tested. It can be seen that the RMSEs decrease continuously with ΔT_s. For $\Delta T_s \gtrsim 25$, the RMSEs are below some threshold values, so the minimally required switching time is about 25 sampling time intervals, which correspond to roughly one cycle of oscillation in the original time series. (To ensure satisfactory performance, we set the switching time ΔT_s to be 100 in all cases.)

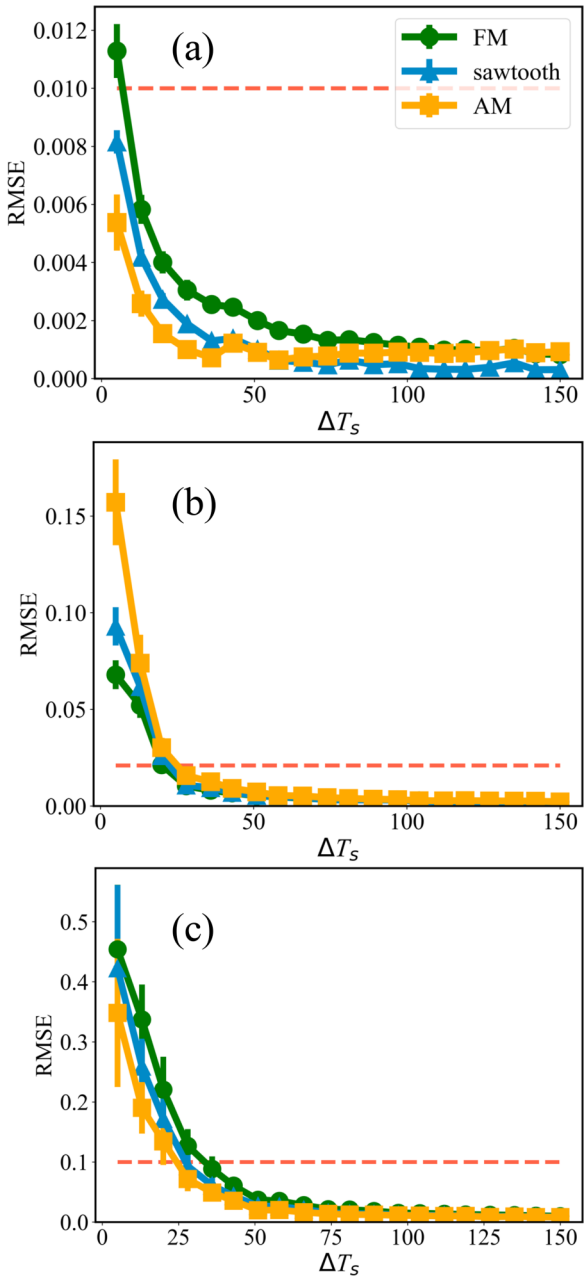

FIG. 8. Determination of the minimum switching time ΔT_s. [(a)–(c)] Testing RMSE versus ΔT_s for tracking the three bifurcation parameters K, y_c, and y_p of the chaotic food chain system, respectively, where ΔT_s is in units of the sampling time interval Δ_s, for three types of parameter variations: FM, sawtooth, and AM. Each data point and the associated error bar are obtained from 50 independent runs. In all cases, the minimally required switching time is about 25 sampling time intervals, corresponding roughly to a single cycle of oscillation in the original time series.

IV. DISCUSSION

In realistic applications, a nonlinear dynamical system is never a closed system but interacts with the environment into which it is deployed. The constant interactions lead to time variations in the system parameters. Tracking how some key bifurcation parameters vary with time enables the possible future behavior of the system to be assessed, but this remains to be a challenging problem in nonlinear dynamics, especially when the system equations are unknown and parameter tracking needs to be carried out in a purely data-driven fashion. In recent years, various machine-learning methods were developed to address this challenge, but with the tacit assumption that the full state observation of the system is available, and the ways the parameters vary with time are needed to be known for training [6–11]. The main contribution of our present work are (1) to relax the full state observation condition, i.e., to develop a machine-learning parameter tracking framework based on partial state observation, (2) only several constant parameter values are necessary for training the framework, while in the testing phase, the parameter varies in real time, and (3) the historical parameter values are not used in real-time parameter tracking. The importance and necessity of relying on partial observation and only a few parameter values for parameter tracking are evident: in applications not all the dynamical variables and parameters of the systems are accessible and there can even be unknown hidden variables.

This paper developed a reservoir-computing based framework to continuously track parameter variations with time based on observation from a subset of the full state variables. Relying on partial observation is necessary in real applications where not all the dynamical variables of the system are accessible and there can be unknown hidden variables. Our unique training scheme enables the reservoir computer to learn the correspondence between the input time series and the underlying parameter value so that it is able to forecast the parameter variations with time based on input time series from partial state observation. We demonstrated the working of the proposed parameter-tracking framework using chaotic systems subject to three distinct types of parameter variations. The effects of a number of factors on the tracking performance were investigated: minimally required bifurcation parameter values for training, the number of state vectors associated with partial-state observation, measurement and dynamical noises, network size and training time, and the minimum switching time required of the training data. The developed reservoir-computing based scheme represents a general and robust parameter-tracking framework that can be deployed in real-world applications.

A key feature of our machine-learning framework is to track the complicated time variations of some inherent parameter of the system, rather than predicting its dynamical state. Our framework is model free and fully data driven: the equations of the system are not needed. There are potential applications. For example, in a power grid where the equations of the whole network are unknown, continuous parameter tracking with time will allow us to detect if the network is under attack by observing the current flows in a subset of transmission lines. In an ecological system, even if the precise relationships among the different spices are unknown, the reproduction rate of some species or environmental factors such as the degree of drought can be estimated and their time variations can be tracked by observing the populations of some species.

In our study, three forms of the time variations of the parameter to be tracked were considered: FM, AM, and sawtooth wave. None of them is linear. Numerically, machine-learning training using time series from three distinct parameter values appears sufficient for the neural network to learn the "dynamical climate" of the underlying system, as done in previous works on predicting critical transitions in and digital twins of nonlinear dynamical systems [30,38], but this phenomenon has not been theoretically understood.

A possible application is in epidemiology. For example, some virus has seasonal behaviors: its spreading rate varies in different seasons. Our framework can be used to track the spreading rate change with time based on state evolution, thereby enabling accurate prediction of the infection scale and period. Another potential application is predicting if a dynamical system is about to approach a tipping point at which a transition from a normal to a collapsing (e.g., large-scale extinction in an ecosystem) steady state occurs. With inevitable noises, we can use the noisy time series from partial state observation as the input to our machine-learning scheme to generate the time-varying behavior of some key bifurcation parameters of the system. Based on the predicted trend of the parameter variation, the chance of a tipping point occurring in the near future can be assessed.

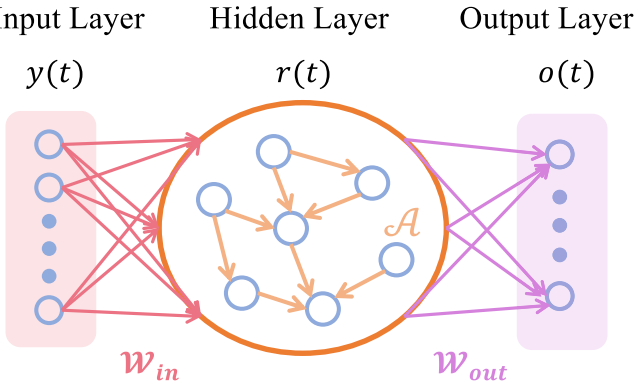

FIG. 9. Basic configuration of a reservoir computer.

ACKNOWLEDGMENT

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

APPENDIX A: RESERVOIR COMPUTING

Reservoir computing (RC) is a class of recurrent neural networks (RNNs) that consists of three layers: an input layer, a hidden recurrent layer, and an output layer, where the input weights and the parameters of the hidden layer network such as the edge weights are defined before training and remain fixed. At the end of the training phase, a linear regression is performed to tune the weights of the output layer, which are characterized by the output matrix $\mathcal{W}_{\text{out}}$. These properties render efficient the training of RC with performance similar to that of the conventional RNNs [61].

Figure 9 illustrates the basic operational scheme of RC: The input matrix $\mathcal{W}_{\text{in}}$ maps the reservoir input signal $\mathbf{y}(t)$, which is typically low dimensional, into the high-dimensional state vector of the hidden layer network with n nodes. Activated by the sequence of reservoir input signals $[\mathbf{y}(:,1), \mathbf{y}(:,2), \ldots, \mathbf{y}(:,t)]$, the hidden layer state $\mathbf{r}(t)$ is updated step-by-step according to

$$\mathbf{r}(t+1) = (1-\alpha)\cdot\mathbf{r}(t) + \alpha\cdot\tanh[\mathcal{A}\cdot\mathbf{r}(t) + \mathcal{W}_{\text{in}}\cdot\mathbf{y}(t) + \mathcal{W}_{\text{bias}}], \quad \text{(A1)}$$

where α is the leaking parameter that determines the rate of "leakage" in reservoir state updating, and the activation is governed by the hyperbolic tangent function (tanh). The bias vector $\mathcal{W}_{\text{bias}}$ inside the tanh function is composed of equal constants w_{b}, whose role is to shift the small signals into the nonlinear region [37]. The quantities D_y, D_r, and D_o are the dimensions of the input signal $\mathbf{y}(t)$, the hidden state $\mathbf{r}(t)$ and the output signal $\mathbf{o}(t)$, respectively. The dimension of the input matrix $\mathcal{W}_{\text{in}}$ is $D_r * D_y$, whose elements are generated uniformly in the range $[-\gamma, \gamma]$ prior to training. The dimension of the hidden layer connection network $\mathcal{A}$ is $D_r * D_r$, whose elements are Gaussian random numbers before training, given the network link probability p and spectral radius ρ. Generally, D_r is much larger than D_y in order to assure

TABLE II. Optimal reservoir-computing hyperparameter values for tracking the variations of the three different parameters of the chaotic Rössler oscillator Eq. (B1).

Bifurcation parameter	ρ	γ	α	β	p
a	2.12	1.62	0.18	$10^{-1.3}$	0.99
b	0.39	3.42	0.19	$10^{-6.3}$	0.03
c	2.39	2.90	0.06	$10^{-1.9}$	0.30

that the reservoir computer is sufficiently complex and high-dimensional to learn dynamical system generating the input signal. The dimension of the output matrix $\mathcal{W}_o$ is $D_o * D_r$. In the training phase, the reservoir state $\mathbf{r}(t)$ is updated step by step according to Eq. (A1) and then concatenated into the matrix $\mathcal{R}$ that has dimension $D_r * T_{\text{train}}$, where T_{train} is the training length. Combined with the input time-series matrix $\mathcal{U}$, the output matrix $\mathcal{W}_{\text{out}}$ can be calculated by using Tikhonov regularization [62]

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

where $\mathcal{I}$ is the identity matrix of dimension D_r, β is the regularization coefficient, and $\mathcal{R}'$ is the transpose of reservoir state matrix $\mathcal{R}$. In the testing phase, with the observed system state $\mathbf{y}(t)$ as the input, the predicted parameter value $\mathbf{o}(t)$ at time t is given by

$$\mathbf{o}(t) = \mathcal{W}_{\text{out}}\mathbf{r}(t). \tag{A3}$$

Hyperparameter optimization is essential for reservoir computing to achieve the desired performance. We apply Bayesian optimization [63] from Matlab (*surrogateopt*) to find the optimal values of the following hyperparameters: the leakage α, the regularization coefficient β, the scaling factor γ of the input matrix $\mathcal{W}_{\text{in}}$, the spectral radius ρ of the recurrent network in the hidden layer, the link probability p determining the network connection matrix $\mathcal{A}$, and the constant in the bias matrix w_b.

APPENDIX B: PARAMETER TRACKING FOR CHAOTIC RÖSSLER OSCILLATOR

The chaotic Rössler system [56] is described by

$$\begin{aligned} \frac{dx}{dt} &= -y - z, \\ \frac{dy}{dt} &= x + ay, \\ \frac{dz}{dt} &= b + z(x - c), \end{aligned} \tag{B1}$$

for $a = 0.2$, $b = 0.2$, and $c = 5.7$. We focus on tracking a single parameter, with the other two fixed. Figure 10 displays some typical bifurcation diagrams of the system.

Table II lists the optimal hyperparameter values for tracking the three parameters: a, b, and c in the chaotic Rössler oscillator Eq. (B1) for different time-varying waveforms. Representative results are presented in Fig. 11, where the shaded regions in the left column indicate the partial observation $\mathbf{y}$ and the three panels in the right column present tracking results with AM, FM, and sawtooth parameter variations,

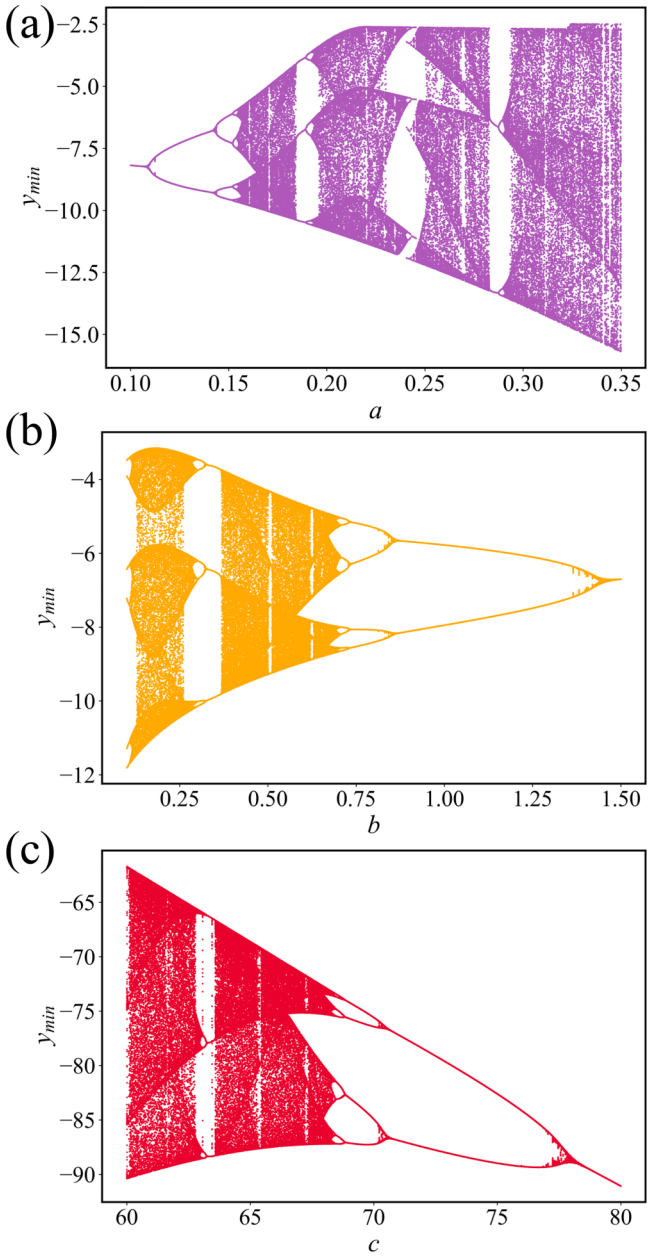

FIG. 10. Bifurcation diagrams of the chaotic Rössler system. [(a)–(c)] Diagrams with respect to bifurcation parameters a, b, and c, respectively.

respectively. In all cases, the sampling number is $s_n = 5$ and the tracking performance is reasonable. Figure 12 shows the testing RMSE versus s_n for the three parameters in the Rössler system and three types of time-varying waveforms. In all cases, it can be seen that choosing $s_n \geqslant 3$ leads to small errors. Figure 13 demonstrates the effect of varying the relative range s_w of the parameter variation from which the training time series are taken on the tracking performance. As s_w decreases from 100%, the RMSE increases slowly, but large error will arise when s_w is larger than 30%.

The effects of various machine-learning characteristics on the parameter-tracking performance for the Rössler oscillator have been studied. First, the system is three-dimensional,

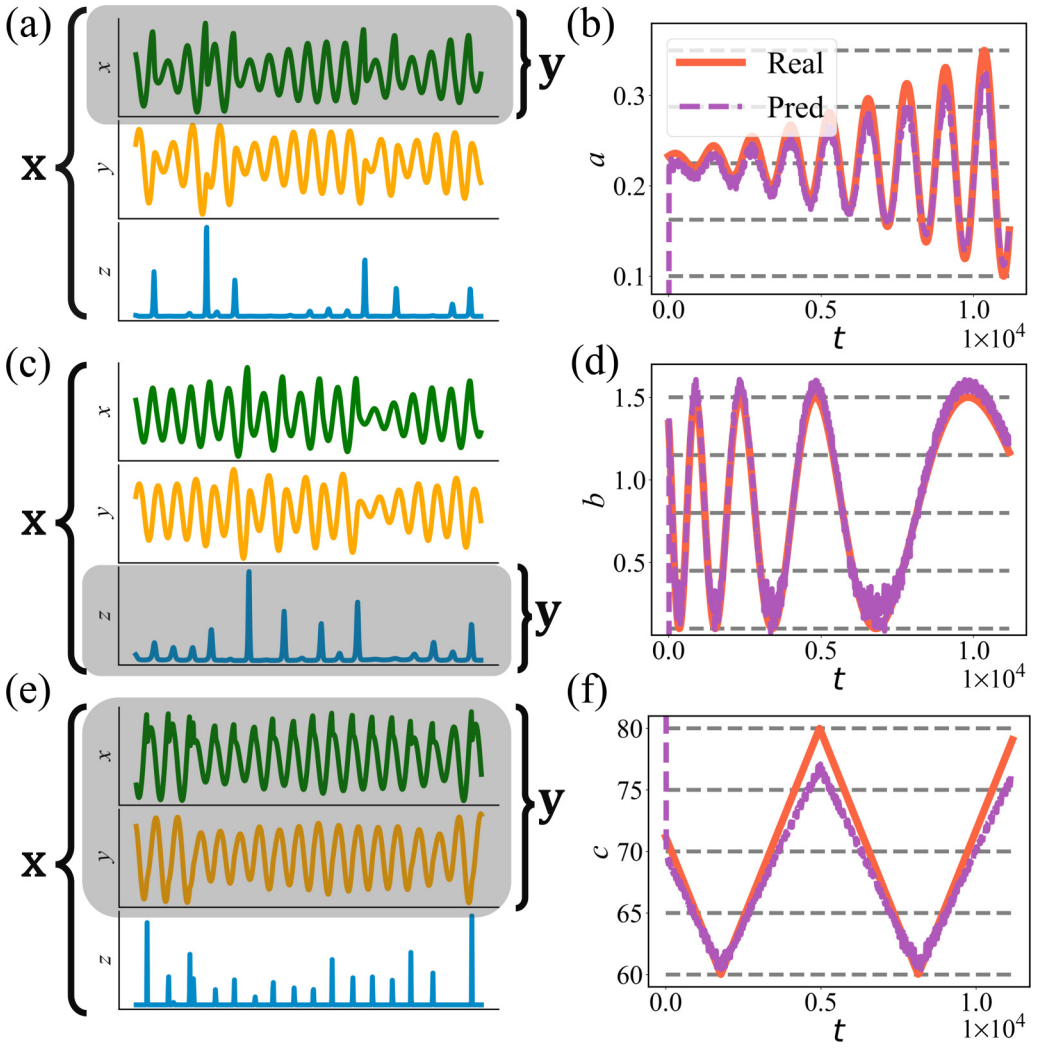

FIG. 11. Tracking time-varying parameters of the chaotic Rössler system based on partial state observation. Different combinations of the parameter-variation waveforms and partial state observation are illustrated: the top, middle, and bottom rows correspond to three types of parameter variations (AM, FM, and sawtooth wave), while the left column illustrates the partial state observation. The full state is $\mathbf{x} = (x, y, z)$ and the partial state observation used for parameter tracking is indicated by the gray shade. The right column presents the results of parameter tracking, where both the machine-learning result and the ground truth are displayed. [(a) and (b)] One-dimensional observational state variable $\mathbf{y} = (x)$ for tracking AM parameter a. [(c) and (d)] Two-dimensional observation $\mathbf{y} = (z)$ for tracking FM parameter b. [(e) and (f)] Two-dimensional observation $\mathbf{y} = (x, y)$ for tracking the sawtooth-wave parameter c. The five horizontal dashed lines indicate the parameter values from which training data are generated: $s_n = 5$.

so there are seven configurations of partial state observation (including full state observation as a special case) (x_1), (x_2), (x_3), (x_1, x_2), (x_1, x_3), (x_2, x_3), (x_1, x_2, x_3). Figure 14 shows the success probability for the three parameters. It can be seen from Figs. 14(a) and 14(b) that, for tracking parameters a and b in Eq. (B1), close to 100% success rate can be achieved by observing two state variables. However, for tracking the third parameter c, it is necessary to observe the first two state variables. Second, we have tested the robustness of parameter tracking against measurement and dynamical noises that are normally distributed with zero mean and standard deviation σ_m and σ_d, respectively. Representative results are shown in Fig. 15. It can be seen that the RMSEs are small and vary slowly with σ_m or σ_d, insofar as these noise amplitudes are below some threshold. Third, the effects of the reservoir network size D_r and training time T_{train} on the tracking performance for the Rössler system are illustrated in Fig. 16. As the network becomes larger, the success probability can be dramatically improved, but the training time has a relatively small effect on the performance. Fourth, the effect of switching time ΔT_s on the tracking performance has been studies, as shown in Fig. 17. For $\Delta T_s \gtrsim 70$, the RMSEs are below some threshold values, so the minimally required switching time is about 70 sampling time intervals, corresponding to roughly two cycles of oscillation in the original time series.

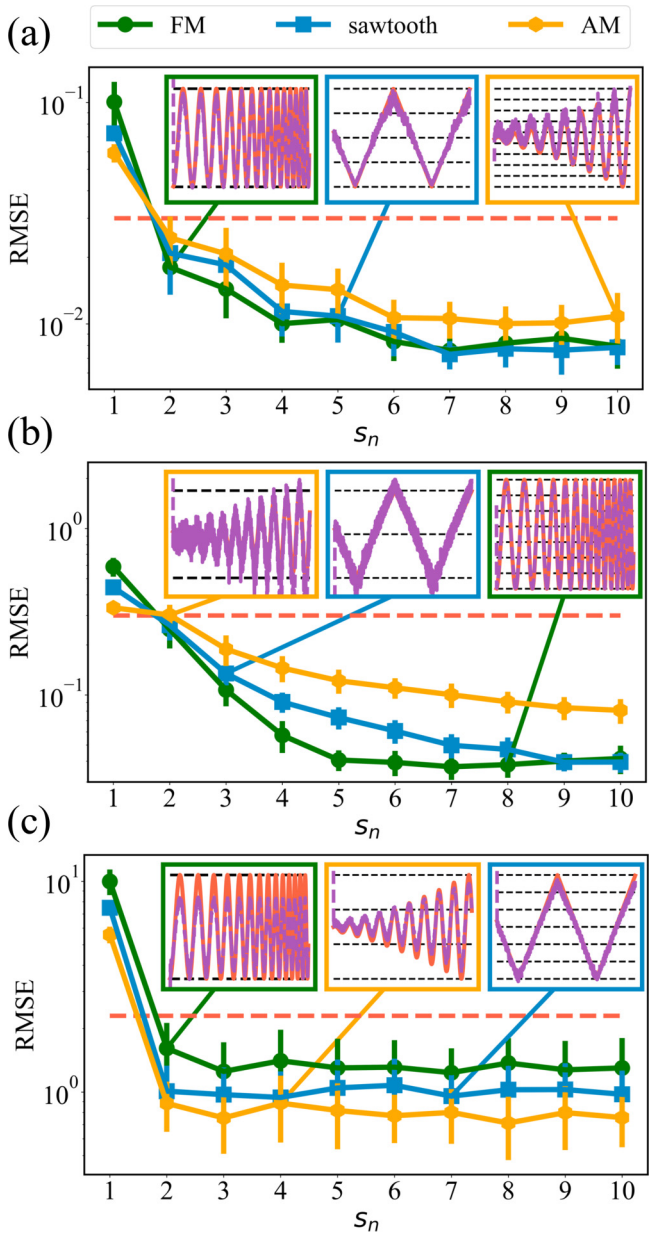

FIG. 12. Determining the number s_n of the distinct values of the bifurcation for training. [(a)–(c)] Testing RMSE versus s_n for the bifurcation parameters a, b, and c in the Rössler system, respectively. Each panel contains results from three types of parameter variations: FM, sawtooth, and AM wave. Each point is the result of averaging over 50 independent runs. The horizontal dashed lines in the insets indicate the specific values of the bifurcation parameter from which the training data are generated. For $s_n = 1$, the testing error is large. As s_n increases, the error decreases. For $s_n \geqslant 3$, the errors become smaller than some threshold, indicating when s_n is above three, accurate parameter tracking can be achieved.

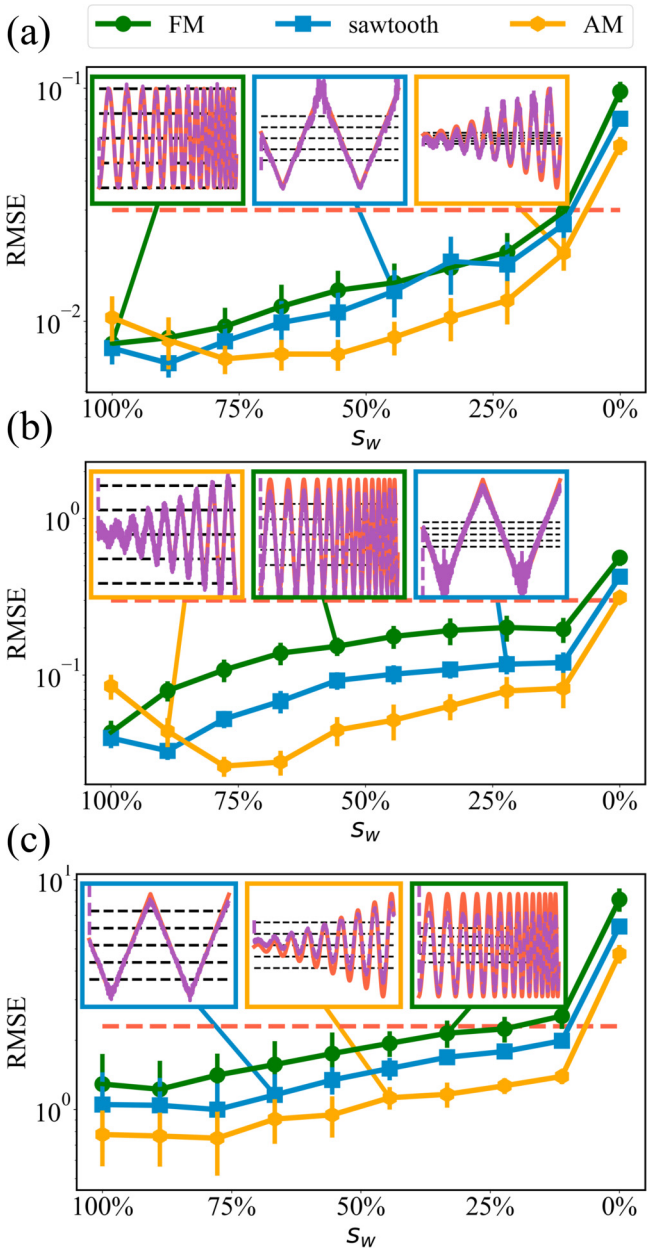

FIG. 13. Effect of the relative range s_w of the time variations of the bifurcation parameter from which time series are taken for training. [(a)–(c)] Testing RMSE versus s_w for the bifurcation parameters a, b, and c in the Rössler system, respectively. Each panel contains results from three types of parameter variations: FM, sawtooth, and AM wave. Each point is the result of averaging over 50 independent runs. The horizontal dashed lines in the insets indicate the specific values of the bifurcation parameter from which the training data are generated. In most cases, the error increases slowly as s_w decreases to about 30%, indicating a high level of tolerance of the machine-learning parameter tracking scheme to the range of the bifurcation parameter in which the training time series are generated.

APPENDIX C: PARAMETER TRACKING FOR MACKEY-GLASS SYSTEM

The Mackey-Glass system is described by a time-delayed nonlinear differential equation [57,64]. Because of the time delay, the system has a memory so it is non-Markovian and in principle has an infinite dimensional phase space. The equation was introduced to model healthy and pathological behaviors in some biological systems [57], which is

$$\frac{dx}{dt} = \frac{ax(t-\tau)}{1+x(t-\tau)^c} - bx(t), \tag{C1}$$

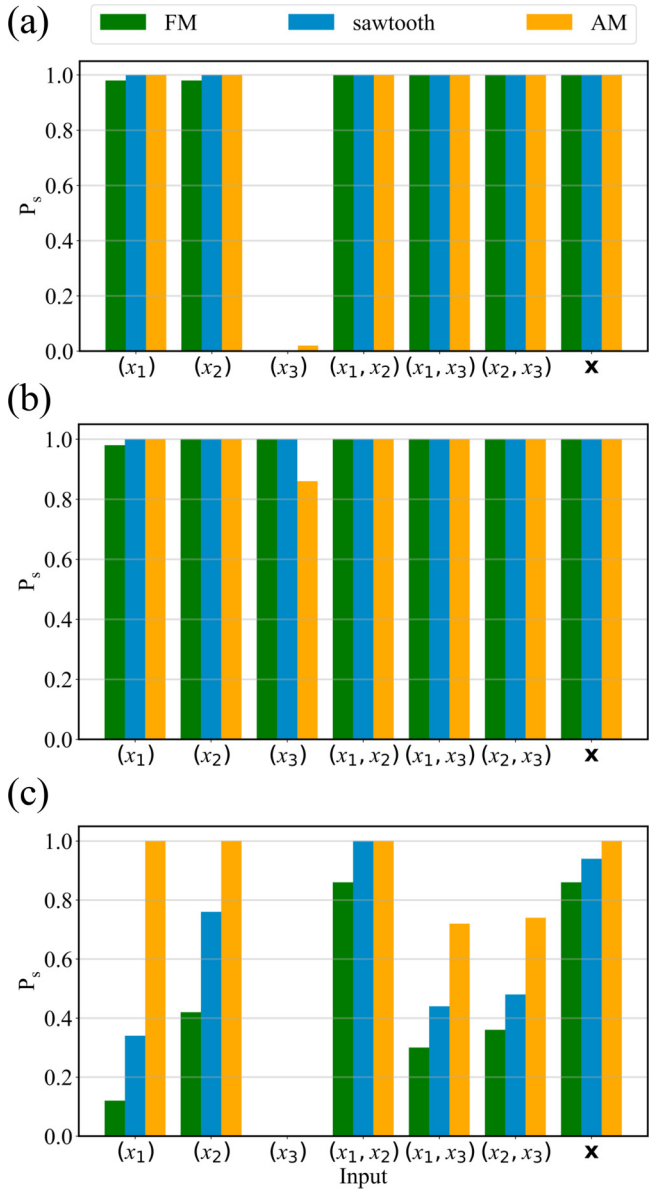

FIG. 14. Success probability for different partial observation scenarios for the chaotic Rössler system. [(a)–(c)] Bar-graph representation of the success rate for the seven distinct cases of state observation for three different bifurcation parameters. Each case contains results from three types of parameter variation (FM: green, sawtooth: blue, AM: yellow). The success rate is calculated from 50 independent realizations.

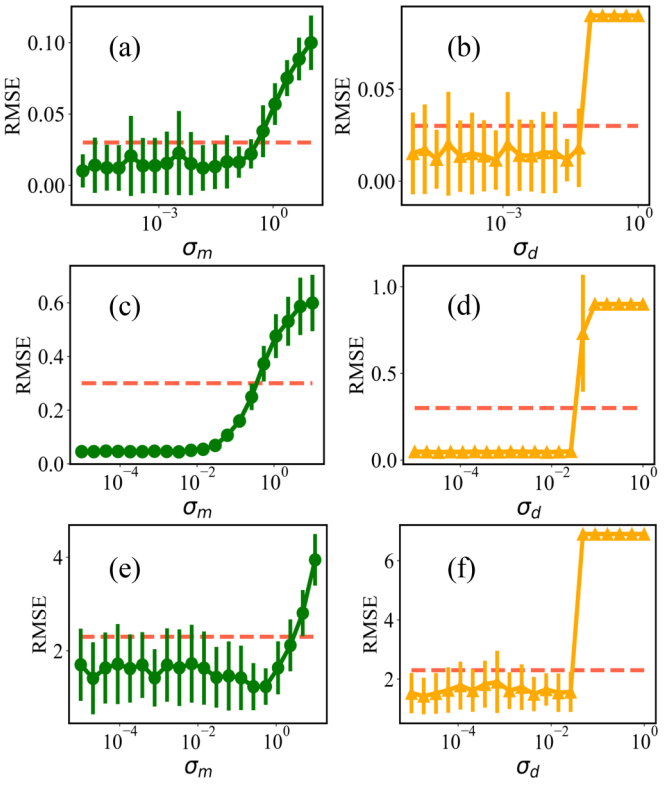

FIG. 15. Robustness of parameter tracking against measurement and dynamical noises. The upper, middle, and lower rows correspond to tracking the three parameters (a, b, and c) in the chaotic Rössler system, respectively. The left and right columns are for measurement and dynamical noises, respectively. The type of parameter variations is used. Each value of testing RMSE and the error bar is obtained from 50 independent training and test runs. The RMSEs remain small and approximately constant if the noise amplitude is below some reasonable value. The horizontal dashed lines are indicative of some (arbitrary) threshold below which the tracking performance is deemed satisfactory.

where τ is the delayed time, a, b, and c are parameters. Depending on the parameter values, distinct dynamical behaviors can arise. For example, for $a = 0.2$, $b = 0.1$, and $c = 10$, the attractor of the system is periodic for $\tau \lesssim 17$ and chaotic for $\tau \gtrsim 17$. Suppose τ is the time-varying parameter to be tracked. To be concrete, we assume that τ varies in the range [14,22]. A representative bifurcation diagram is shown in Fig. 18.

For fixed $a = 0.2$, $b = 0.1$, and $c = 10$, the reservoir-computing hyperparameters for tracking the time-varying parameter τ are determined to be $\rho = 1.26$, $\gamma = 0.73$, $\alpha = 0.31$, $\beta = 10^{-6.9}$, and $p = 0.17$. Figures 19(a) and 19(b) show the testing RMSE versus s_n and s_w, respectively, for FM, sawtooth, and AM types of parameter variations. It can be seen from Fig. 19(a) that, $s_n = 2$ is sufficient for tracking the time variation of τ accurately, and Fig. 19(b) demonstrates that the subrange of the parameter variation from which the training data are taken can be as low as 10% while still generating small RMSEs. Robustness of tracking the time variation of τ against measurement and dynamical noises is illustrated in Fig. 20. The effects of reservoir-computing network size D_r, training time T_{train}, and switching time ΔT_s on parameter-tracking performance is exemplified in Fig. 21.

It is worth noting that while the chaotic Mackey-Glass system contains a single dynamical variable, the phase-space dimension is infinite. If the time-delay parameter τ is a constant, then it can be estimated by calculating the linear autocorrelation. However, in our study, τ is time varying, which cannot be revealed by linear autocorrelation.

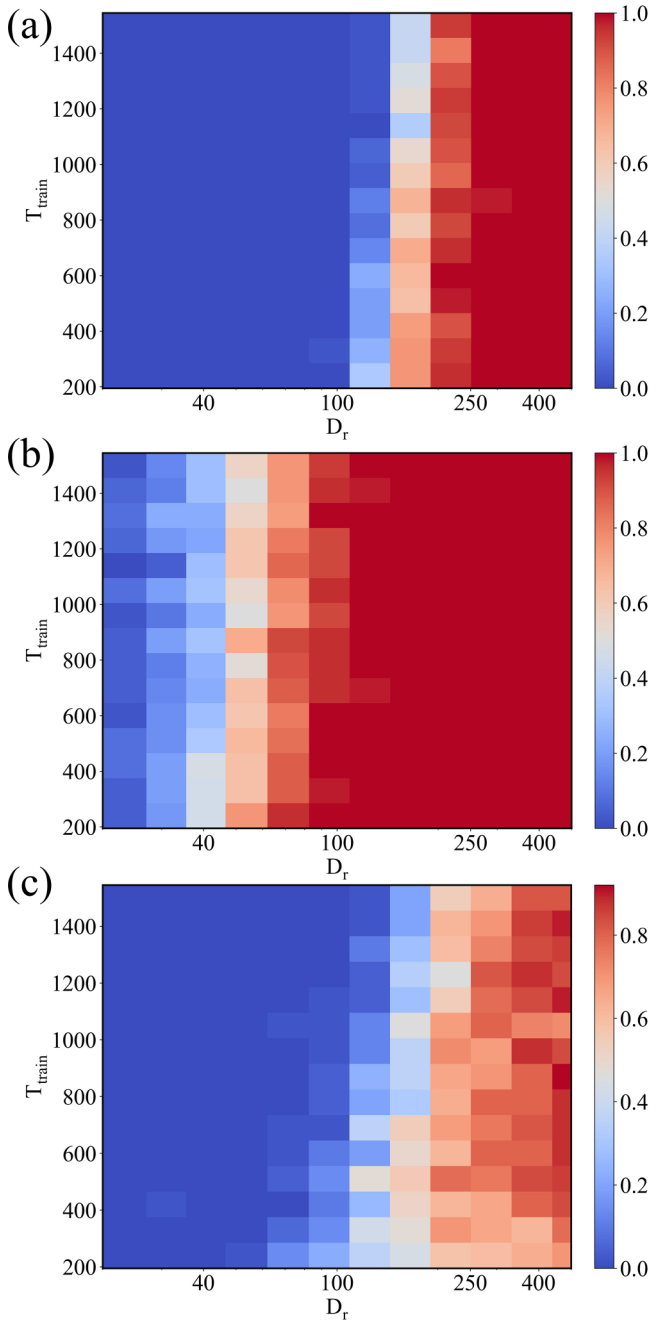

FIG. 16. Effects of two hyperparameters: the network size D_r and training time T_{train}, on parameter-tracking performance. [(a)–(c)] Color-coded success probability values in the plane of these two hyperparameters for tracking the bifurcation parameters a, b, and c in the Rössler system, respectively, subject to FM type of variations. Each value of the success probability is obtained using 50 statistical realizations. Increasing the network size can often dramatically improve the success rate.

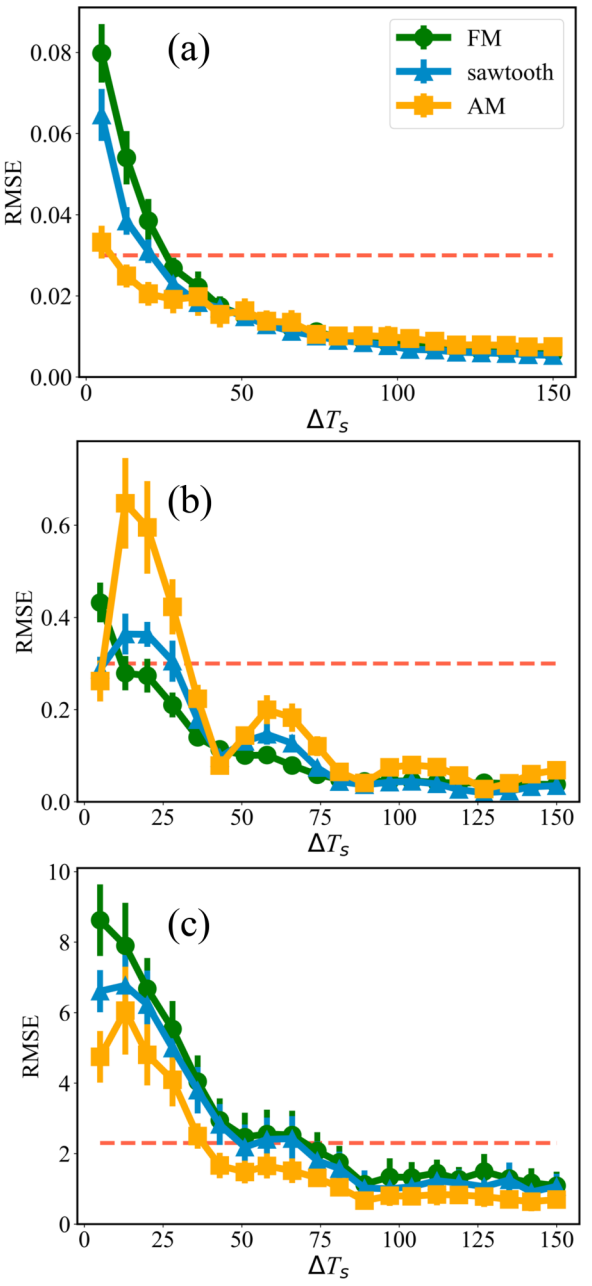

FIG. 17. Determination of the minimum switching time ΔT_s. [(a)–(c)] Testing RMSE versus ΔT_s for tracking the three bifurcation parameters a, b, and c of the chaotic Rössler system, respectively, where ΔT_s is in units of sampling time interval Δ_s. In each case, three types of parameter variations are tested: FM, sawtooth, and AM. Each data point and the associated error bar are obtained from 50 independent runs. In all cases, the minimally required switching time is about 70 sampling time intervals, corresponding approximately to two cycles of oscillation in the original time series.

APPENDIX D: TRACKING COMPLEX PARAMETER VARIATIONS FOR THE LORENZ-96 SYSTEM

The Lorenz-96 model [58] is a benchmark system in meteorological and climatological studies, which is capable of replicating complex behaviors akin to those observed in real atmospheric systems. The model is essentially a system of N coupled oscillators, which described by the following set of N coupled, first-order differential equations:

$$\frac{dx_j}{dt} = -x_{j-1}(x_{j-2} - x_{j+1}) - x_j + F(t), \qquad \text{(D1)}$$

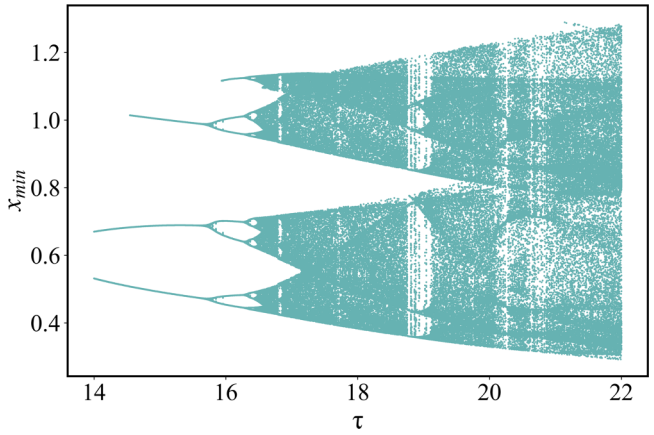

FIG. 18. A representative bifurcation diagram the Mackey-Glass system with the time delay τ being the bifurcation parameter.

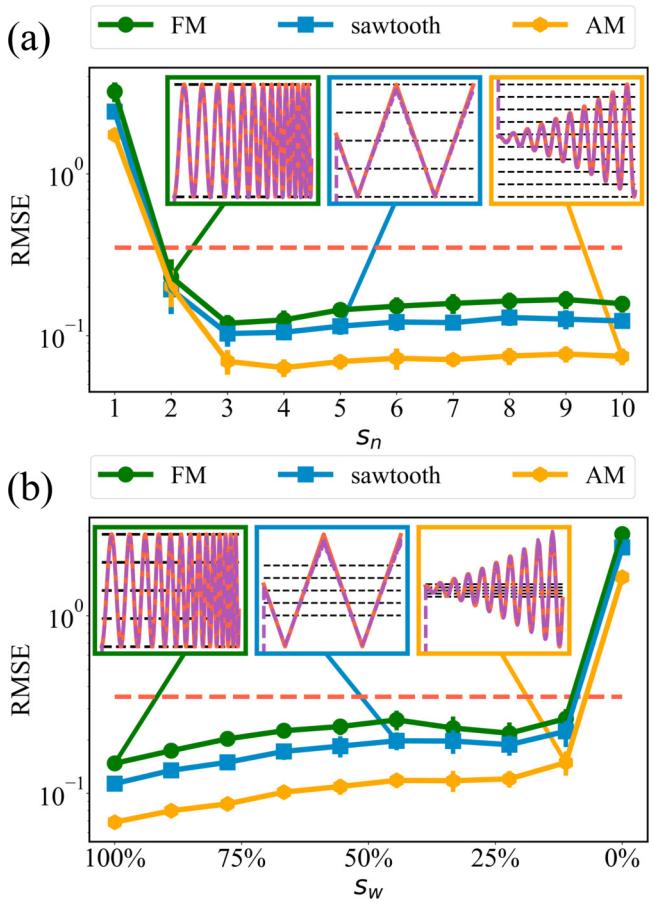

FIG. 19. Demonstration of parameter-tracking performance for the Mackey-Glass system. The time delay τ is the parameter whose time variation is to be tracked. The other three parameters in Eq. (C1) are fixed: $a = 0.2$, $b = 0.1$, and $c = 10$. [(a) and (b)] Testing RMSEs versus s_n and s_w, respectively, for three different types of parameter variation: FM, sawtooth, and AM. Panel (a) indicates that using time series from three distinct values of the bifurcation parameter suffices for accurate parameter tracking. Panel (b), the error increases slowly as s_w decreases to about 20%, indicating a high level of tolerance of the scheme to the range of the bifurcation parameter in which the training time series are generated.

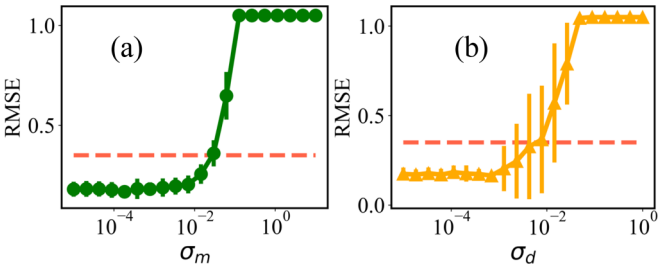

FIG. 20. Robustness against noise for tracking the time-delay parameter in the chaotic Mackey-Glass system. [(a) and (b)] RMSE versus σ_m and σ_d, the amplitudes of the measurement and dynamical noise, respectively, for the FM type of parameter variations. Each value of testing RMSE and the error bar is obtained from 50 independent training and test runs. The RMSEs remain small and approximately constant if the noise amplitude is below some reasonable value. The horizontal dashed lines are indicative of some (arbitrary) threshold below which the tracking performance is deemed satisfactory.

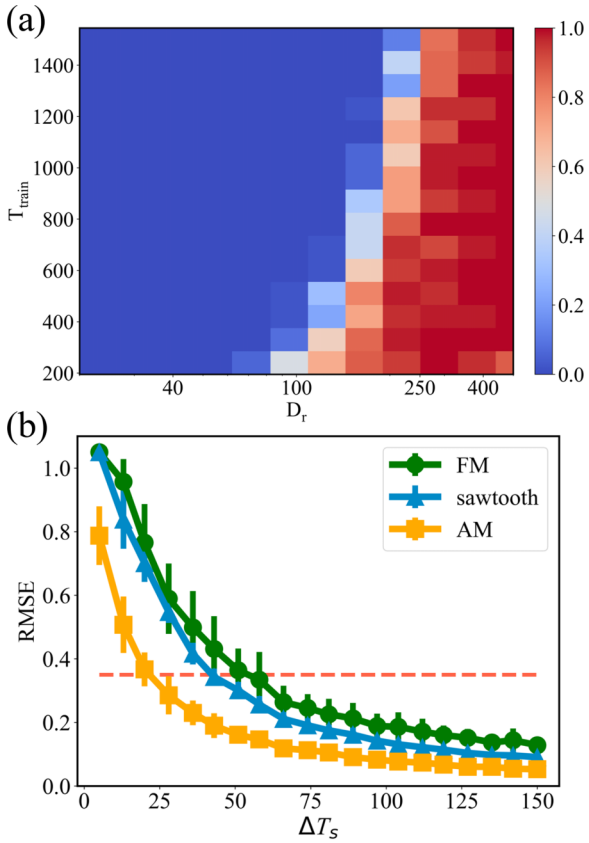

FIG. 21. Effects of reservoir-computing network size D_r, training time T_{train}, and the switching time ΔT_s on parameter-tracking performance for the Mackey-Glass system. (a) Color-coded success probability values in the plane of these two hyperparameters for tracking the time-delay parameter τ, subject to FM type of variations. (b) Testing RMSE versus ΔT_s for three types of parameter variations: FM, sawtooth, and AM. Each value of the success probability is obtained using 50 statistical realizations. Increasing the network size and the training time can often dramatically improve the success rate. In all cases, the minimally required switching time is about 60 sampling time intervals, corresponding roughly to three cycles of oscillation in the original time series.

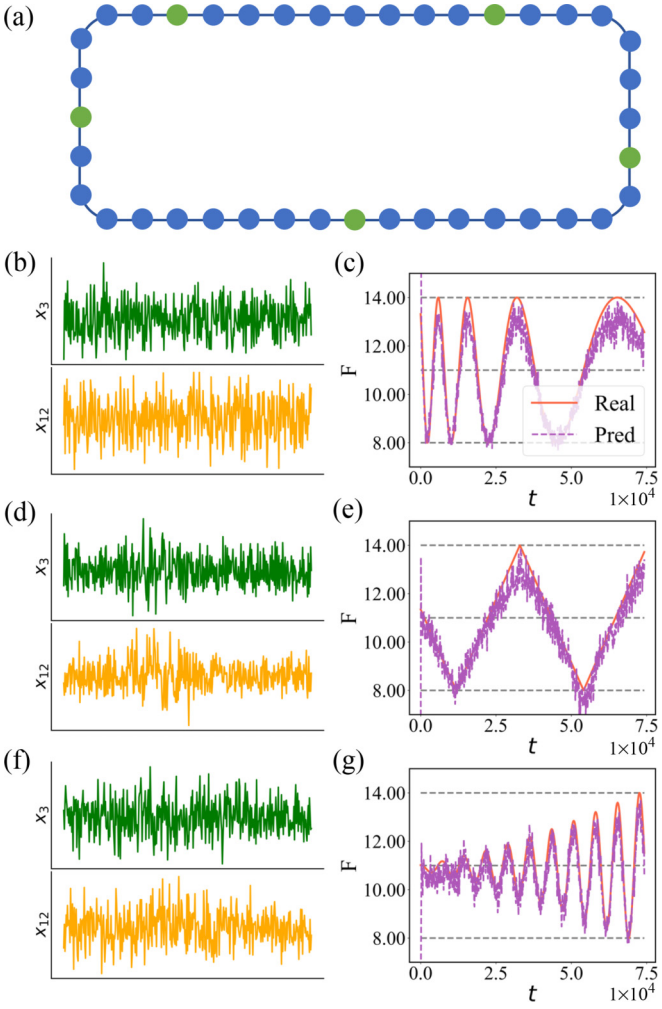

FIG. 22. Tracking time-varying parameters of the chaotic Lorenz-96 system based on partial state observation. (a) The system is 40-dimensional and the observation is made on five sites. [(b) and (c)] Representative observed time series and tracking of FM parameter variations, respectively. [(d) and (e)] Same as in (b) and (c) but for AM parameter variations. [(f) and (g)] Same as in (b) and (c) but for sawtooth parameter variations. In (c), (e), and (g), the three horizontal dashed lines indicate the parameter values from which training data are obtained: $s_n = 3$.

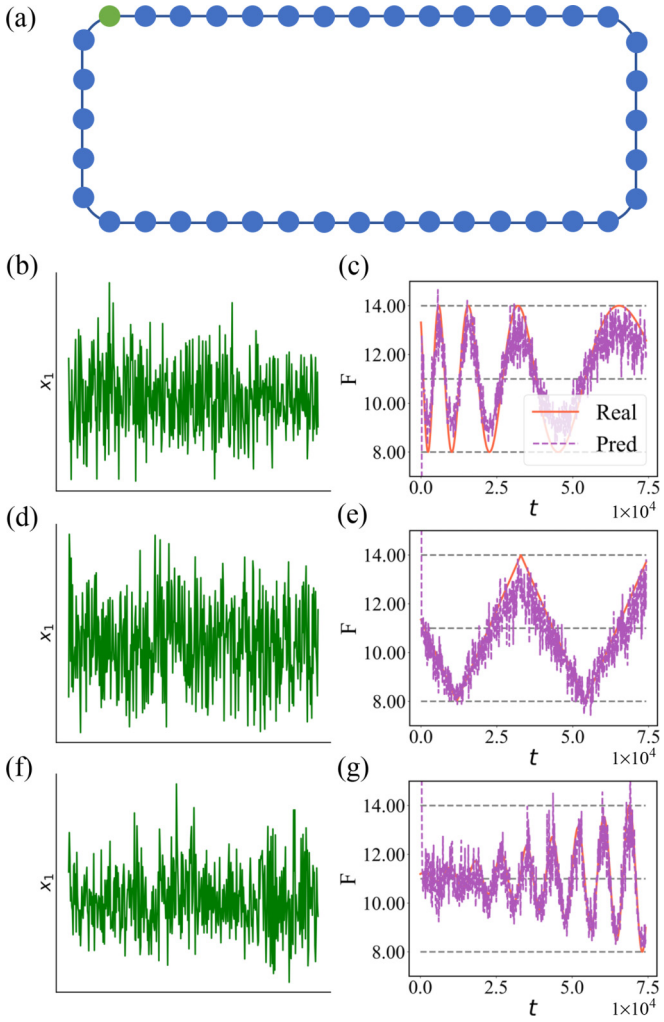

FIG. 23. Tracking time-varying parameters of the 40-dimensional chaotic Lorenz-96 system with a single measurement. Legends are the same as in Fig. 22 but for observing a single state variable as indicated by the green dot in (a).

where x_j is the state variable at position j, for $j = 1, \ldots, N$, and $F(t)$ is a forcing term. To be concrete, we set $N = 40$, regard $F(t)$ as the time-varying parameter to be tracked in the range [8, 14], and assume periodic boundary conditions so topologically the oscillators are located on a ring with nearest-neighbor couplings. The 40-dimensional chaotic Lorenz-96 system constitutes a spatiotemporal nonlinear system.

The reservoir-computing hyperparameters for tracking the time-varying parameter $F(t)$ are determined to be $\rho = 2.95$, $\gamma = 0.17$, $\alpha = 0.23$, $\beta = 10^{-5.6}$, and $p = 0.56$. Figures 22 and 23 show the parameter-tracking results where the partial state observation is taken from five and one random locations, respectively, as indicated by the green dots in the top panel. In both figures, the left column displays some representative time-series observation and the right column shows the corresponding parameter-tracking results, for three types of time variations (FM, AM, and sawtooth wave). It can be seen that our machine-learning framework is able to track the parameter variations with quite limited observation. It is worth noting that the parameter structure of the Lorenz-96 system is relatively simple: the forcing is additive and uniformly applied to all the oscillators.

We have also attempted to track the parameter variations for the spatiotemporal system described by the Kuramoto-Sivashinsky equation: $u_t + \nu u_{xxxx} + \phi(u_{xx} + uu_x) = 0$, where $u(x, t)$ is a scalar field, ν and ϕ are the system parameters. Note that, in this case, the two parameters are no longer additive to the equation but are multiplicative as they are directly multiplied to terms that involve the scale field and its derivatives. Our extensive and exhaustive numerical computations failed to yield any satisfactory results of parameter tracking. The failure represents a limitation of our machine-learning

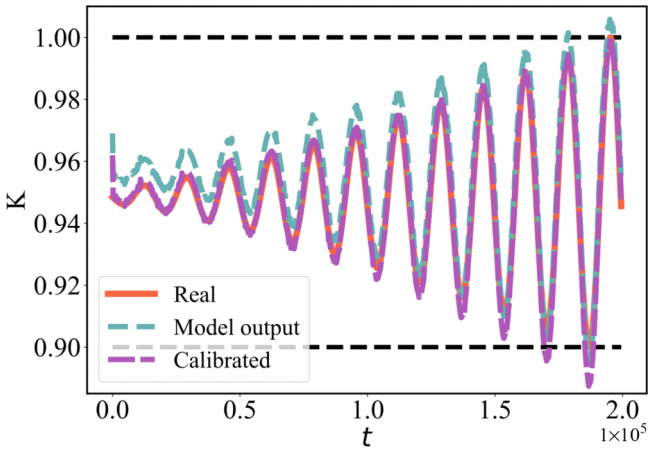

FIG. 24. Illustration of the calibration method. The process is to remove the constant bias from the machine-learning predicted parameter variation.

based parameter tracking framework, warranting further efforts into the development of a more applicable scheme for tracking parameter variations in spatiotemporal dynamical systems.

APPENDIX E: CALIBRATION METHOD

To remove the constant bias between the machine predicted parameter variation and the ground truth, we assume that two true values of the parameter are available for calibration. The calibrated parameter variation is obtained by subtracting the constant bias from the machine-learning output. As an example, we consider the case of tracing the time-varying parameter K in the food chain system Eq. (1). Figure 24 illustrates that, after removing the constant bias, the calibrated machine-learning prediction of the parameter variation (green dashed curve) agrees well with the ground truth (solid dark-red curve).

[1] C. Grebogi, E. Ott, and J. A. Yorke, Crises, sudden changes in chaotic attractors and chaotic transients, Physica D **7**, 181 (1983).

[2] Y.-C. Lai and T. Tél, *Transient Chaos—Complex Dynamics on Finite Time Scales*, 1st ed. (Springer, New York, 2011).

[3] Z.-M. Zhai, M. Moradi, L.-W. Kong, B. Glaz, M. Haile, and Y.-C. Lai, Model-free tracking control of complex dynamical trajectories with machine learning, Nat. Commun. **14**, 5698 (2023).

[4] A. Wikner, J. Pathak, B. R. Hunt, I. Szunyogh, M. Girvan, and E. Ott, Using data assimilation to train a hybrid forecast system that combines machine-learning and knowledge-based components, Chaos **31**, 053114 (2021).

[5] A. Farchi, P. Laloyaux, M. Bonavita, and M. Bocquet, Using machine learning to correct model error in data assimilation and forecast applications, Quart. J. Roy. Meteorol. Soc. **147**, 3067 (2021).

[6] J. Hannink, T. Kautz, C. F. Pasluosta, K.-G. Gaßmann, J. Klucken, and B. M. Eskofier, Sensor-based gait parameter extraction with deep convolutional neural networks, IEEE J. Biomed. Health Info. **21**, 85 (2016).

[7] Y. Chen and Y. Zhou, Machine learning based decision making for time varying systems: Parameter estimation and performance optimization, Knowl. Bas. Sys. **190**, 105479 (2020).

[8] X. Chen, Y. Tian, T. Zhang, and J. Gao, Differential evolution based manifold gaussian process machine learning for microwave filter's parameter extraction, IEEE Access **8**, 146450 (2020).

[9] Y. Zhang, Neural network algorithm with reinforcement learning for parameters extraction of photovoltaic models, IEEE Trans. Neural Netw. Learn. Syst. **34**, 2806 (2021).

[10] A. B. Abdusalomov, F. Safarov, M. Rakhimov, B. Turaev, and T. K. Whangbo, Improved feature parameter extraction from speech signals using machine learning algorithm, Sensors **22**, 8122 (2022).

[11] M.-Y. Kao, F. Chavez, S. Khandelwal, and C. Hu, Deep learning-based BSIM-CMG parameter extraction for 10-nm FinFET, IEEE Trans. Electr. Devices **69**, 4765 (2022).

[12] H. Jaeger, The "Echo State" Approach to Analysing and Training Recurrent Neural Networks—With an Erratum Note, German National Research Center for Information Technology GMD Technical Report, Bonn, Germany (2001).

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

[14] M. Lukoševičius and H. Jaeger, Reservoir computing approaches to recurrent neural network training, Comput. Sci. Rev. **3**, 127 (2009).

[15] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, Information processing using a single dynamical node as complex system, Nat. Commun. **2**, 468 (2011).

[16] 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).

[17] 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).

[18] 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).

[19] 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).

[20] 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).

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

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

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

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

[25] 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).

[26] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, Recent advances in physical reservoir computing: A review, Neural Netw. **115**, 100 (2019).

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

[28] C. Zhang, J. Jiang, S.-X. Qu, and Y.-C. Lai, Predicting phase and sensing phase coherence in chaotic systems with machine learning, Chaos **30**, 083114 (2020).

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

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

[31] D. Patel, D. Canaday, M. Girvan, A. Pomerance, and E. Ott, Using machine learning to predict statistical properties of non-stationary dynamical processes: System climate, regime transitions, and the effect of stochasticity, Chaos **31**, 033149 (2021).

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

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

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

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

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

[37] T. L. Carroll, Optimizing memory in reservoir computers, Chaos **32**, 023123 (2022).

[38] L.-W. Kong, Y. Weng, B. Glaz, M. Haile, and Y.-C. Lai, Reservoir computing as digital twins for nonlinear dynamical systems, Chaos **33**, 033111 (2023).

[39] J. Z. Kim and D. S. Bassett, A neural machine code and programming framework for the reservoir computer, Nat. Mach. Intell. **5**, 622 (2023).

[40] Z.-M. Zhai, M. Moradi, L.-W. Kong, and Y.-C. Lai, Detecting weak physical signal from noise: A machine-learning approach with applications to magnetic-anomaly-guided navigation, Phys. Rev. Appl. **19**, 034030 (2023).

[41] E. W. Weisstein, Least squares fitting (2002), https://mathworld.wolfram.com/.

[42] J.-X. Pan and K.-T. Fang, Maximum likelihood estimation, in *Growth Curve Models and Statistical Diagnostics* (Springer, Berlin, 2002), pp. 77–158.

[43] A. J. Haug, *Bayesian Estimation and Tracking: A Practical Guide* (John Wiley & Sons, New York, 2012).

[44] L. Yao and W. A. Sethares, Nonlinear parameter estimation via the genetic algorithm, IEEE Trans. Signal Process. **42**, 927 (1994).

[45] P. Guo, M. R. Lyu, and C. L. P. Chen, Regularization parameter estimation for feedforward neural networks, IEEE Trans. Syst. Man, Cybern. B **33**, 35 (2003).

[46] F. Yandun, M. Torres-Torriti, and F. Auat Cheein, Markov chain monte carlo parameter estimation for nonzero slip models of wheeled mobile robots: A skid-steer case study, J. Mech. Robot. **13**, 050902 (2021).

[47] R. Van Der Merwe and E. A. Wan, The square-root unscented Kalman filter for state and parameter-estimation, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (IEEE, Los Alamitos, CA, 2001), Vol. 6, pp. 3461–3464.

[48] G. Evensen, The ensemble Kalman filter for combined state and parameter estimation, IEEE Contr. Syst. Mag. **29**, 83 (2009).

[49] T. A. Wenzel, K. Burnham, M. Blundell, and R. Williams, Dual extended Kalman filter for vehicle state and parameter estimation, Vehicle Syst. Dyn. **44**, 153 (2006).

[50] H. Moradkhani, S. Sorooshian, H. V. Gupta, and P. R. Houser, Dual state–parameter estimation of hydrological models using ensemble Kalman filter, Adv. Water Resour. **28**, 135 (2005).

[51] L. Ljung, Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems, IEEE Trans. Autom. Control **24**, 36 (1979).

[52] B. A. Acuña Acurio, D. E. C. Barragán, J. C. L. Amezquita, M. J. Rider, and L. C. P. D. Silva, Design and implementation of a machine learning state estimation model for unobservable microgrids, IEEE Access **10**, 123387 (2022).

[53] P. Litkey, X. Liang, H. Kaartinen, J. Hyyppä, A. Kukko, M. Holopainen, R. Hill, J. Rosette, and J. Suárez, Single-scan TLS methods for forest parameter retrieval, Proc. SilviLaser **2008**, 8th (2008).

[54] D. Musicki, R. Evans, and B. La Scala, Multi-scan parametric target tracking in clutter, in *Proceedings of the 42nd IEEE International Conference on Decision and Control* (IEEE, Los Alamitos, CA, 2003), Vol. 5, pp. 5372–5377.

[55] K. McCann and P. Yodzis, Nonlinear dynamics and population disappearances, Am. Natural. **144**, 873 (1994).

[56] O. E. Rössler, An equation for continuous chaos, Phys. Lett. A **57**, 397 (1976).

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

[58] E. N. Lorenz, Predictability: A problem partly solved, in *Proceedings of the Seminar on Predictability* (ECMWF, Reading, UK, 1996), Vol. 1.

[59] F. Takens, Detecting strange attractors in turbulence, in *Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a Symposium Held at the University of Warwick 1979/80* (Springer, Berlin, 2006), pp. 366–381.

[60] C. Van den Broeck, J. M. R. Parrondo, R. Toral, and R. Kawai, Nonequilibrium phase transitions induced by multiplicative noise, Phys. Rev. E **55**, 4084 (1997).

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

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

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

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