



## OPEN ACCESS

EDITED BY  
Sou Nobukawa,  
Chiba Institute of Technology, Japan

REVIEWED BY  
Teruya Yamanishi,  
Osaka Seikei University, Japan  
Teijiro Isokawa,  
University of Hyogo, Japan

\*CORRESPONDENCE  
Keita Tokuda  
✉ k.tokuda.jm@juntendo.ac.jp

RECEIVED 20 June 2024  
ACCEPTED 30 September 2024  
PUBLISHED 22 October 2024

CITATION  
Tokuda K and Katori Y (2024) Prediction of unobserved bifurcation by unsupervised extraction of slowly time-varying system parameter dynamics from time series using reservoir computing.  
*Front. Artif. Intell.* 7:1451926.  
doi: 10.3389/frai.2024.1451926

COPYRIGHT  
© 2024 Tokuda and Katori. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Prediction of unobserved bifurcation by unsupervised extraction of slowly time-varying system parameter dynamics from time series using reservoir computing

Keita Tokuda<sup>1\*</sup> and Yuichi Katori<sup>2</sup>

<sup>1</sup>Faculty of Health Data Science, Juntendo University, Urayasu, Japan, <sup>2</sup>The School of Systems Information Science, Future University Hakodate, Hakodate, Japan

**Introduction:** Nonlinear and non-stationary processes are prevalent in various natural and physical phenomena, where system dynamics can change qualitatively due to bifurcation phenomena. Machine learning methods have advanced our ability to learn and predict such systems from observed time series data. However, predicting the behavior of systems with temporal parameter variations without knowledge of true parameter values remains a significant challenge.

**Methods:** This study uses reservoir computing framework to address this problem by unsupervised extraction of slowly varying system parameters from time series data. We propose a model architecture consisting of a slow reservoir with long timescale internal dynamics and a fast reservoir with short timescale dynamics. The slow reservoir extracts the temporal variation of system parameters, which are then used to predict unknown bifurcations in the fast dynamics.

**Results:** Through experiments on chaotic dynamical systems, our proposed model successfully extracted slowly varying system parameters and predicted bifurcations that were not included in the training data. The model demonstrated robust predictive performance, showing that the reservoir computing framework can handle nonlinear, non-stationary systems without prior knowledge of the system's true parameters.

**Discussion:** Our approach shows potential for applications in fields such as neuroscience, material science, and weather prediction, where slow dynamics influencing qualitative changes are often unobservable.

## KEYWORDS

chaos, nonlinear dynamics, bifurcation (mathematics), reservoir computing, slow - fast dynamics

## 1 Introduction

Nonlinear, non-stationary processes are abundant in various natural and physical phenomena. For instance, the dynamics of neurons are known to be strongly dependent on the state of the brain, determined by varying levels of attention, arousal, anesthesia, and sleep depth, as well as on different behavioral patterns such as movement (Steriade et al., 1993; Buzski, 2002; Tokuda et al., 2019; Vohryzek et al., 2020). Similarly, the response of physical systems can qualitatively change due to bifurcation phenomena as

sample properties or experimental conditions vary (Bnard, 1901; Ertl, 1991; Itoh and Kimoto, 1996; Raab et al., 2023). Various mathematical frameworks have been proposed to model non-stationary dynamics (Waddington, 1961; Kaneko and Tsuda, 2003; Rabinovich et al., 2001; Katori et al., 2011; Patel et al., 2021). One plausible and simple depiction is that system parameters vary over time or in different contexts (Patel et al., 2021).

Consider either a discrete nonlinear dynamical system:

$$\mathbf{x}(n+1) = f(\mathbf{x}(n); \lambda), \quad (1)$$

or a continuous dynamical system:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}; \lambda), \quad (2)$$

where  $\mathbf{x} \in \mathbb{R}^n$  represents the dynamical variable expressing fast dynamics, and  $\lambda$  is a parameter of function  $f$  whose value can potentially lead to bifurcation in the dynamics of  $\mathbf{x}$ . In the context of modeling static nonlinear systems with a fixed value of  $\lambda$ , recent advancements in machine learning have enabled the rules governing the underlying system to be extracted and learned from observed time series data with much higher accuracy than before. In particular, by learning from time series data, reservoir computing has facilitated the creation of autonomous dynamical systems within the model that can generate time series resembling those of the target system, achieving high accuracy even in challenging problems such as learning chaotic systems. Furthermore, recent studies have demonstrated the prediction of unobserved bifurcations that are not present in the learning data (Kong et al., 2021; Patel et al., 2021; Kim et al., 2021; Itoh, 2023). In their settings, they have succeeded in predicting unknown bifurcations that occur when the parameter  $\lambda$  takes values other than those used when generating the observed data. For example, Patel et al. addressed the bifurcation parameter  $\lambda$  of a chaotic dynamical system not as static value but as a variable changing very slowly over time, and learned the time series generated by this system. After learning the one-step-ahead prediction task, they added a feedback loop to the reservoir, creating a closed-loop model that can generate time series as an autonomous dynamical system. They showed that, although learning the time series of  $x$  using the conventional reservoir computing framework alone does not predict unobserved bifurcations, successful learning can be achieved by separately providing the reservoir with the true value of the parameter at each moment as an additional input. When the parameter values inputted during the prediction phase were different from those during learning, the model was able to predict bifurcations not included in the training data. Kim et al. demonstrated that the emergence of a Lorenz attractor not present in the training data could be predicted by first inputting time series generated from the Lorenz equations along with the true bifurcation parameter values into the reservoir, then forming a closed-loop model to create an autonomous dynamical system, and finally changing the input parameter values. These studies indicate that predicting unknown bifurcation phenomena is possible by additionally inputting the value of the bifurcation parameter into the reservoir. This suggests that the reservoir computing framework is capable of learning not just specific dynamical systems but families of dynamical systems, suggesting the potential

to predict the emergence of system states qualitatively different from those observed in real data. However, these prior studies assume that the true value of the parameter is known, which is not the case in most real-world scenarios, including in brain data observation. Therefore, the question arises whether the behavior of non-stationary systems with temporal parameter variations can be predicted solely from observed time series data.

Various methods, including recurrence plots (Marwan and Kraemer, 2023), supervised learning (Zhai et al., 2024), slow feature analysis (Wiskott and Sejnowski, 2002; Antonelo and Schrauwen, 2012), and hierarchical structures (Yonemura and Katori, 2021; Katori, 2019; Gallicchio et al., 2017; Tamura et al., 2019), have been reported for extracting the slowly moving components of system dynamics. In this study, we leverage the reservoir computing framework to address this problem. Our central idea is based on the following consideration: in a typical scenario, a reservoir receives a signal derived from a nonlinear dynamical system, such as one variable of the state vector  $\mathbf{x}$  — e.g.,  $x_1$  —, in one step and predicts its value in the next time step. Previous studies have indicated that establishing generalized synchronization between the reservoir and the original system generating the input signal is crucial for achieving accurate predictions (Rulkov et al., 1995; Carroll, 2020; Lu et al., 2018; Lu and Bassett, 2020), where generalized synchronization refers to the condition that the listening reservoir's state,  $\mathbf{u}(t)$ , is a continuous function,  $\Psi(\mathbf{x})$ , of the state of the original system,  $\mathbf{x}$ . Especially, if the function  $\Psi(\mathbf{x})$  is invertible, the reservoir's state  $\mathbf{u}(t)$  has all the information about  $\mathbf{x}$ . It is reasonable to predict the value of another element — e.g.,  $x_2$  —, from partial observation of the system — e.g., only  $x_1$  —, if generalized synchronization is established between the original system and the reservoir (Lu et al., 2017). Now, considering the parameter  $\lambda$  varies slowly over time as expressed in Equation 2, the following system can be formulated:

$$\begin{cases} \frac{d\mathbf{x}}{dt} = f_{\mathbf{x}}(\mathbf{x}; \lambda) \\ \frac{d\lambda}{dt} = f_{\lambda}(\mathbf{x}, \lambda) \end{cases} \quad (3)$$

Let  $X$  be a concatenation of  $\mathbf{x}$  and  $\lambda$ , defined as  $X = {}^t(\mathbf{x}, \lambda)$ , then this system can be represented as a single ordinary differential equation (ODE):

$$\frac{dX}{dt} = F(X). \quad (4)$$

We assume that the signal is generated from the trajectory of this concatenated system's attractor. When the signal originating from  $\mathbf{x}$  is input into the reservoir and invertible generalized synchronization between the reservoir state  $\mathbf{u}$  and  $X$  is achieved, the reservoir's state has full information about  $\lambda$ . While above discussion is speculative, previous studies have shown that by adjusting the reservoir's timescale and structure, the reservoir can successfully extract the slow dynamics of the signal source system (Manneschi et al., 2021; Jaeger, 2008; Gallicchio et al., 2017; Tanaka et al., 2022; Yonemura and Katori, 2021). The extraction of such slow or static system states within the reservoir computing framework, where internal couplings are not altered during learning, suggests that unsupervised extraction of such information is possible using reservoirs. We first aim to verify

whether it is possible to extract the true variation of parameter  $\lambda$ 's by simply extracting the slowly varying variables within the reservoir (Section 3.1 Experiment 1).

Patel et al. (2021) have demonstrated that predicting the time series of the concatenated system  $X$  cannot be achieved by a simple single reservoir. The challenges addressed in this paper are twofold: (1) estimating the unobservable slowly varying parameter values (Section 3.1 Experiment 1), and (2) predicting unknown bifurcations in the fast dynamics under the variation of such parameters (section 3.2 Experiment 2). While the second challenge has been tackled by Patel et al. and Kim et al. in scenarios where the true parameter value is known, in this study, we explore the possibility of learning from observational data generated by nonlinear systems and predicting unknown bifurcations without the knowledge of true parameter values. We allow the bifurcation parameter values to change over time but assume these changes occur on a significantly longer timescale compared to the system's fast dynamics. Previous studies suggest that extracting the slowly changing parameter values from time series observations in an unsupervised manner may allow us to substitute the true parameter value with an estimated one.

The architecture of the model proposed in this study comprises two types of reservoirs stacked in layers: a slow reservoir with long timescale internal dynamics and a fast reservoir with short timescale dynamics. Assuming a nonlinear system with a very slowly changing bifurcation parameter value as the signal source, we input observational data obtained from the fast dynamics into these reservoirs. We found that when the variables that change slowly are extracted from the internal state of the slow reservoir, they trace the temporal variation of the system's parameter. Although the variables extracted from the slow reservoir differ in amplitude scale from the true parameter values, we show that adding these variables and the observational time series to the fast reservoir allows for the prediction of unknown bifurcations, resembling the true parameter values provided in prior studies.

## 2 Materials and methods

### 2.1 Problem setting

Consider the following nonlinear differential equations:

$$\begin{cases} \frac{d\mathbf{x}}{dt} = f_{\mathbf{x}}(\mathbf{x}; \lambda) \\ \frac{d\lambda}{dt} = f_{\lambda}(\lambda, t) \end{cases} \quad (5)$$

As implied by this equation, the parameter  $\lambda$  is assumed to vary over time. However, in this paper, we sometimes do not explicitly define  $f_{\lambda}$  and, instead, assume  $\lambda$  is simply a function of  $t$ . In either case, the temporal change of  $\lambda$  is assumed to be significantly slower than that of  $\mathbf{x}$ . Herein, we consider concrete examples of  $f_{\mathbf{x}}$  by examining numerical computations derived from the Lorenz and Rössler equations.

We assume the observation time series  $y$  is given as a function of the fast variable  $\mathbf{x}$ , as follows:

$$y(n) = g(\mathbf{x}(\Delta t \cdot n)) \quad (6)$$

where  $g(\mathbf{x}(t))$  is a function of  $\mathbf{x}(t)$ . We assume  $g$  to be well-behaved, such as a smooth and differentiable function, but without including the full observation of the state  $\mathbf{x}$ , i.e.,  $\dim y < \dim \mathbf{x}$ . In this paper, we use

$$g(\mathbf{x}) = x_1, \quad (7)$$

where  $x_1$  is the first element of the vector  $\mathbf{x}$ . We naturally suppose that the time series are obtained by temporally discretizing the continuous signal with a specific time step,  $\Delta t$ . The problem we address here is whether detecting changes in the slowly varying parameter  $\lambda$  and predicting unknown bifurcations is possible based on the observed  $y(n)$ .

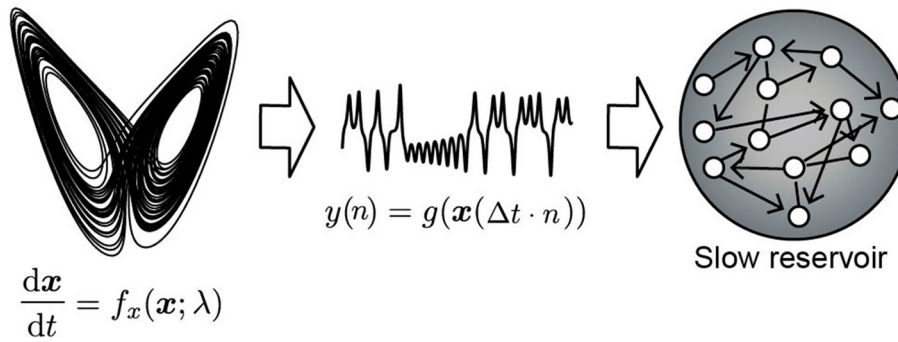
### 2.2 Model architecture

In this study, we conduct two experiments: (1) estimating the unobservable slowly varying parameter values (Section 3.1 Experiment 1), and (2) predicting unknown bifurcations in the fast dynamics under the variation of such parameters (Section 3.2 Experiment 2). In Experiment 1, as shown in Figure 1, we input time series observations generated from the attractor trajectory of a nonlinear dynamical system into a reservoir with a slow time constant and observe the internal state of the reservoir. We check whether there are nodes in the internal state that exhibit fluctuations similar to the slow movement of the parameter of the dynamical system generating the data. The experimental setup is shown in Figure 1. We refer to this reservoir with a slow time constant as the *slow reservoir*. In Experiment 2, we test whether the movements of the parameters extracted from the slow reservoir and the observed time series can be used as inputs to predict bifurcations in the attractor. We refer to this downstream reservoir as the *fast reservoir*. The model architecture is shown in Figure 2. In Experiment 2, during the training phase, the model learns from the time series, and afterwards, by introducing feedback, it operates as a fully autonomous system. The model must predict both the changes in the slow-moving parameter and the values of the fast dynamical variables. Therefore, the model includes a third reservoir, called the *slow dynamics predictor*, which predicts the time series output of the slow reservoir (Figure 2). In the test phase after training, both the slow dynamics predictor and the fast reservoir are provided with feedback from their outputs, forming a *closed-loop model*.

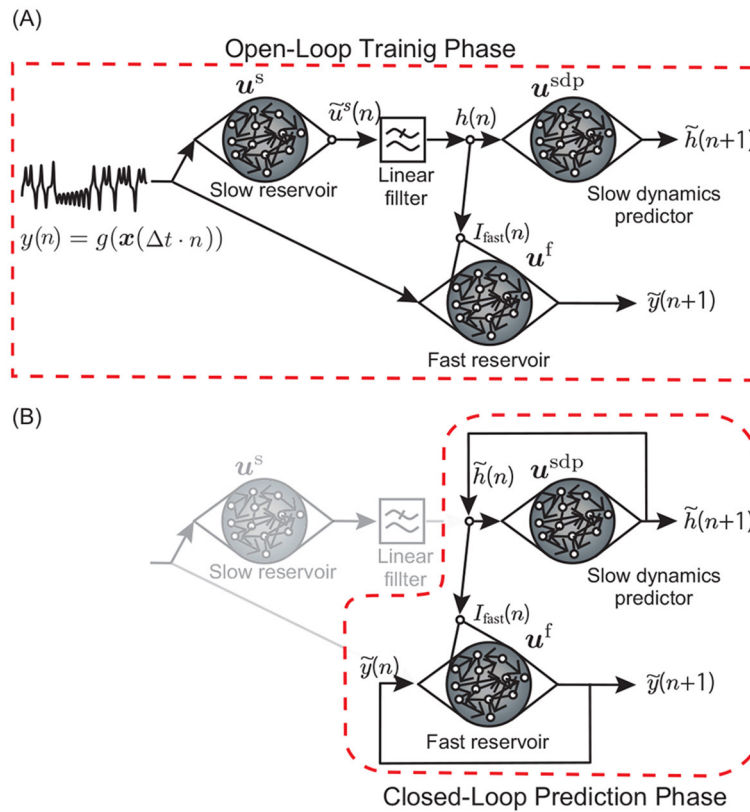
### 2.3 Reservoir model

#### 2.3.1 Slow reservoir

We employed a reservoir consisting of leaky integrator neurons with long time constants to extract the slow variables of the system (Jaeger et al., 2007). Namely, the leak rate,  $\alpha$ , is set close to 1. To ensure that the reservoir dynamics arising from neuron interactions also exhibit long time constants, we adjust the spectral radius of the recurrent connection strength,  $\mathbf{W}^s \in \mathbb{R}^{N^s \times N^s}$  to be equal to one. The dynamics of the slow reservoir are defined by the



**FIGURE 1** Schematic diagram of the numerical examination discussed in Section 3.1. The observation signal  $y(n)$  is generated by a nonlinear system  $dx/dt = f_x(x; \lambda)$ . The slow reservoir consists of leaky neuron models with a very long leak rate, and the spectral radius of the recurrent connection is set to 1.



**FIGURE 2** Schematic diagram showing the open-loop training phase and the closed-loop prediction phase. **(A)** In the training phase, observation  $y(n)$  is fed to both the slow and fast reservoirs. The output of the slow reservoir is also input to the fast reservoir. Additionally, the output of the slow reservoir is input to the “slow dynamics predictor” reservoir. After the training phase, the output weights of both the slow dynamics predictor and the fast reservoir are optimized to conduct one-step-ahead prediction of their own inputs. **(B)** In the prediction phase, feedback loops are added to the slow dynamics predictor and the fast reservoir to make the whole system a single autonomous dynamical system that can predict time series of  $y(n)$ .

following equation:

$$u^s(n+1) = \alpha u^s(n) + (1 - \alpha) \tanh(W^s u^s(n) + W_{in}^s \cdot y(n) + b^s) \quad (8)$$

where  $u^s = (u_1^s, u_2^s, \dots, u_{N^s}^s)^T$  is the internal state of the reservoir,  $b^s \in \mathbb{R}^{N^s}$  is the bias term,  $W^s \in \mathbb{R}^{N^s \times N^s}$  is the recurrent

connection strength,  $W_{in}^s \in \mathbb{R}^{N^s \times 1}$  is the input connection strength,  $\alpha$  is the leak rate,  $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$  is the hyperbolic tangent,  $N^s$  is the number of neurons in the reservoir, and  $y(n) \in \mathbb{R}$  is a one-dimensional time series observation derived from the data generation models mentioned earlier. The elements of  $W^s$  are drawn from an i.i.d. Gaussian normal distribution, and then  $W^s$  is normalized by multiplying a constant factor so that the spectral

radius  $\rho(W^s)$  satisfies  $\rho(W^s) = 1$ . The elements in the input connection matrix  $W_{in}^s$  and in vector  $b^s$  are drawn from an i.i.d. uniform distribution over intervals  $[-\chi_{in}^s, \chi_{in}^s]$  and  $[-\chi_b^s, \chi_b^s]$ , respectively. The parameter values used are  $N^s = 500$  and  $\alpha = 0.995$ . When applying the Lorenz system as the data generation model, we used  $\chi_{in}^s = 0.5$  and  $\chi_b^s = 5$ , whereas when applying the Rössler model, we set  $\chi_{in}^s = 15$  and  $\chi_b^s = 150$ . To determine the slow dynamics of the target system, we extracted slowly changing elements of the internal state  $u^s$ , which were heuristically selected using the following procedure:

- For each  $i$ , calculate a moving average of the time series  $u_i^s(n)$  using a time window  $n_{window}$  with a specific width, where  $u_i^s(n)$  is the  $i$ th element of the internal state  $u^s$ . Let  $\bar{u}_i^s(n)$  denote this moving average:

$$\bar{u}_i^s(n) = \frac{\sum_{k=0}^{n_{window}-1} u_i^s(n-k)}{n_{window}} \quad (9)$$

- For each  $i$ , calculate the fluctuation around its own moving average during the training phase using the standard deviation  $SD[u_i^s(n) - \bar{u}_i^s(n)]$ .
- Choose the elements with the lowest fluctuation. In this study, the top 10%, namely 50 nodes, were selected. Let  $S_{slow}^{10\%} = \{i \mid u_i^s \text{ is in the slowest 10\% of nodes}\}$  denote this set.
- Calculate the instantaneous average of the absolute values of the selected elements:

$$\tilde{u}^s(n) = \frac{1}{|S_{slow}^{10\%}|} \sum_{i \in S_{slow}^{10\%}} |u_i^s(n)|, \quad (10)$$

where  $\tilde{u}^s(n)$  denotes the extracted slow feature, and  $|S_{slow}^{10\%}|$  is the cardinality of the set  $S_{slow}^{10\%}$ . The absolute value of each node is taken because some nodes exhibit changes that follow the same pattern as the parameter changes, while others show changes that are the inverse. Simply averaging these values would cancel them out, resulting in a near-zero  $\tilde{u}^s(n)$ . By taking the absolute value before averaging, we ensure that the contributions of all selected nodes are positively accounted for, avoiding this cancellation effect due to the central symmetry of the tanh function and the distribution of each weight element.

In the numerical experiment with the closed-loop model discussed in Section 3.2, the extracted slow feature is further smoothed by a linear filter before being fed to the downstream reservoirs (namely, fast reservoir and the slow dynamics predictor) to stabilize the learning process. The filtering is described by the following linear dynamics:

$$h(n+1) = \left(1 - \frac{1}{\tau_f}\right)h(n) + \frac{\tilde{u}^s(n)}{\tau_f}, \quad (11)$$

where  $\tau_f$  is the time constant of the filter. This equation is derived from the following linear dynamics:

$$\tau_f \frac{h(n+\Delta n) - h(n)}{\Delta n} = -h(n) + \tilde{u}^s(n) \quad (12)$$

Equation 11 is obtained by substituting  $\Delta n = 1$ . The parameter value used is  $\tau_f = 200$ . The application of a linear filter does not significantly alter the shape of the time series; it is used solely for removing high-frequency components and smoothing (Supplementary Figure S1).

### 2.3.2 Fast reservoir

In addition to the slow reservoir described above, which is employed to extract the slow components of the dynamics, we utilize another reservoir to capture the evolution laws of the fast dynamics of the target system (Figure 2). The model is almost same as Equation 8 but with two inputs and different parameter values:

$$u^f(n+1) = \alpha u^f(n) + (1-\alpha) \tanh(W^f u^f(n) + W_{in}^f \cdot y(n) + W_{param} \cdot I_{fast}(n) + b^f) \quad (13)$$

where  $u^f = (u_1, u_2, \dots, u_{N^f})^T$  is the internal state of the reservoir,  $b^f \in \mathbb{R}^{N^f}$  is the bias term,  $W^f \in \mathbb{R}^{N^f \times N^f}$  is the recurrent connection strength,  $W_{in}^f \in \mathbb{R}^{N^f \times 1}$  is the input connection strength,  $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$  is the hyperbolic tangent, and  $N^f$  is the number of neurons in the reservoir. As in previous studies (Patel et al., 2021; Kim et al., 2021), a slowly changing parameter value that acts as the bifurcation parameter is also fed to the reservoir, as expressed by the term  $W_{param} \cdot I_{fast}(n)$  in the RHS of the equation, where  $W_{param} \in \mathbb{R}^{N^f \times 1}$  is the input connection strength and  $I_{fast}(n)$  is the additional input to the fast reservoir receiving the slow component. Unlike previous studies,  $I_{fast}(n)$  is not the true parameter value of the target system but the output of either one of other reservoirs, the slow reservoir or the slow dynamics predictor described below (Figure 2). A sparse matrix is used for  $W^f$ , such that randomly chosen 2% of the edges are assigned non-zero values, whereas the rest are set to zero. The weight values of the 2% edges are drawn from an i.i.d. uniform distribution over the interval  $[0, 1]$ , and  $W^f$  is normalized so that the spectral radius  $\rho(W^f)$  satisfies  $\rho(W^f) = 0.95$ . The elements in  $W_{in}^f$ ,  $W_{param}$ , and  $b^f$  are drawn from an i.i.d. uniform distribution over intervals  $[-\chi_{in}^f, \chi_{in}^f]$ ,  $[-\chi_{param}^f, \chi_{param}^f]$ , and  $[-\chi_b^f, \chi_b^f]$ , respectively. The parameter values used are  $N^f = 2000$ ,  $\chi_{in}^f = 0.75$ ,  $\chi_{param}^f = 0.15$ ,  $\chi_b^f = 15$ , and  $\alpha = 0.8$ .

### 2.3.3 Slow dynamics predictor

In Section 3.2, we demonstrate the construction of a closed-loop model capable of predicting unobserved bifurcations without requiring an observation signal as its input (Figure 2). Typically, in reservoir computing, a closed-loop model can be established by simply adding a feedback loop, using the reservoir's output as its input at the next time step. This approach is applied to the fast reservoir (Figure 2). However, due to the extraction of slow dynamics from the observation of fast dynamics using the slow reservoir, the output of the slow reservoir exhibits different temporal properties and cannot be used as feedback to substitute the input at the next time step. Therefore, to construct a closed-loop model, we introduce the slow dynamics predictor, which is an additional reservoir that predicts the evolution of the slow component of the target dynamics. In the training phase, the slow

component  $I_{\text{fast}}(n)$  in the RHS of Equation 13 originates from the output of the slow reservoir, namely,  $I_{\text{fast}}(n) = h(n)$ . In the prediction phase, the slow dynamics predictor works as an autonomous system by adding the closed loop, and we substitute  $I_{\text{fast}}(n)$  in the RHS of Equation 13 with the prediction of this slow dynamics predictor, namely,  $I_{\text{fast}}(n) = \tilde{h}(n)$ . We use a standard echo state network for this reservoir (Jaeger and Haas, 2004). The model is almost the same as Equation 13 but with slightly different parameter values and without the input  $y(n)$  and leak term. The dynamics of the slow dynamics predictor during the training phase is described as follows:

$$\begin{aligned} & \mathbf{u}^{\text{sdp}}(n+1) \\ &= \tanh(\mathbf{W}^{\text{sdp}} \mathbf{u}^{\text{sdp}}(n) + \mathbf{W}_{\text{param}} \cdot h(n) + \mathbf{b}^{\text{sdp}}). \end{aligned} \quad (14)$$

The parameter values used are  $N^{\text{sdp}} = 500$  and  $\chi_{\text{param}}^{\text{sdp}} = \chi_{\text{b}}^{\text{sdp}} = 5 \times 10^{-3}$ .

## 2.4 Training

Figure 2 describes the model architecture used for the one-step-ahead prediction task. Training is conducted using “teacher forcing,” where, during the training phase, the observation time series to be predicted serves as the external force that drives the reservoir (Figure 2A). To train the fast reservoir, the sum of squared output errors in one-ahead-prediction and the regularization term is minimized with respect to  $\mathbf{W}_{\text{out}}^f$ :

$$\sum_n \left( y(n+1) - \left( \mathbf{W}_{\text{out}}^f \mathbf{u}^f(n+1) + b_{\text{out}}^f \right) \right)^2 + \beta \|\mathbf{W}_{\text{out}}^f\|_{\text{fro}} \quad (15)$$

where  $\|\mathbf{W}_{\text{out}}^f\|_{\text{fro}}$  is the Frobenius norm of matrix  $\mathbf{W}_{\text{out}}^f$ ,  $\beta$  is the regularization coefficient,  $\mathbf{W}_{\text{out}}^f$  is the output connection strength. Here, the Frobenius norm  $\|A\|_{\text{fro}}$  of a matrix  $A = (a_{ij})$  refers to the square root of the sum of the absolute squares of its elements as follows:

$$\|A\|_{\text{fro}} = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}. \quad (16)$$

The analytic solution for this minimization is given by the following calculation:

$$\begin{pmatrix} \mathbf{W}_{\text{out}} \\ b_{\text{out}} \end{pmatrix} = (\mathbf{U}\mathbf{U}^T + \beta\mathbf{I})^{-1} \mathbf{U}\mathbf{y}, \quad (17)$$

where the left hand side of the equation represents the concatenation of  $\mathbf{W}_{\text{out}}$  and  $b_{\text{out}}^f$  in a vertical manner,  $\mathbf{y} = (y(2), y(3), \dots, y(n+1))^T$  is the vector of observation time series, and

$$\mathbf{U} = \begin{pmatrix} \mathbf{u}^f(1) & \mathbf{u}^f(2) & \dots & \mathbf{u}^f(n) \\ 1 & 1 & \dots & 1 \end{pmatrix} \quad (18)$$

is the matrix whose  $i$ th column is the vertical concatenation of the internal state of the reservoir the input vector at time  $i$ ,  $\mathbf{u}^f(i)$ , and 1.

Similarly, to train the slow dynamics predictor, the sum of squared output errors in one-ahead-prediction and the regularization term is minimized with respect to  $\mathbf{W}_{\text{out}}^{\text{sdp}}$ :

$$\sum_n \left( h(n+1) - \left( \mathbf{W}_{\text{out}}^{\text{sdp}} \mathbf{u}^{\text{sdp}}(n+1) + b_{\text{out}}^{\text{sdp}} \right) \right)^2 + \beta \|\mathbf{W}_{\text{out}}^{\text{sdp}}\|_{\text{fro}}. \quad (19)$$

where  $b_{\text{out}}^{\text{sdp}}$  is the scalar bias term.

The training of the fast reservoir and the slow dynamics predictor take place parallel after the training phase.

## 2.5 Closed-loop model

Following the training phase, feedback loops are incorporated into the slow dynamics predictor and the fast reservoir, transforming the entire system into a single autonomous dynamical system capable of generating predictions for the time series  $y(n)$  (Figure 2B). The fast reservoir with the feedback loop is described by the following equations:

$$\begin{cases} \mathbf{u}^f(n+1) = \tanh(\mathbf{W}^f \mathbf{u}^f(n) + \mathbf{W}_{\text{in}}^f \cdot \tilde{y}^f(n) + \mathbf{W}_{\text{param}} \cdot \tilde{h}(n) + \mathbf{b}^f) \\ \tilde{y}^f(n+1) = \mathbf{W}_{\text{out}}^f \mathbf{u}^f(n+1) + b_{\text{out}}^f \end{cases} \quad (20)$$

where  $\tilde{h}(n)$  is the external input whose evolution is governed by the slow dynamics predictor model described as follows:

$$\begin{cases} \mathbf{u}^{\text{sdp}}(n+1) = \tanh(\mathbf{W}^{\text{sdp}} \mathbf{u}^{\text{sdp}}(n) + \mathbf{W}_{\text{param}} \cdot \tilde{h}(n) + \mathbf{b}^{\text{sdp}}) \\ \tilde{h}(n+1) = \mathbf{W}_{\text{out}}^{\text{sdp}} \mathbf{u}^{\text{sdp}}(n+1) + b_{\text{out}}^{\text{sdp}} \end{cases} \quad (21)$$

Equations 20, 21 collectively form the autonomous dynamical system capable of independently generating a time series.

## 2.6 Largest Lyapunov exponent estimation

After the training phase, the largest Lyapunov exponent (LLE) is computed for the fast reservoir with feedback (Equation 20). In Experiment 2, both during the training and prediction phases, the fast reservoir receives the time-varying output, namely the smoothed output of the slow reservoir,  $h(n)$ , or the output of the slow dynamics predictor,  $\tilde{h}(n)$ , as its input  $I_{\text{fast}}(n)$ . Here we calculate the LLE of the fast reservoir by fixing the value of  $I_{\text{fast}}(n)$ . Namely, the LLE of the fast reservoir with the parameter  $n$  is defined by the LLE of the following dynamical system:

$$\begin{cases} \mathbf{u}^f(k+1) = \tanh(\mathbf{W}^f \mathbf{u}^f(k) + \mathbf{W}_{\text{in}}^f \cdot \tilde{y}^f(k) + \mathbf{W}_{\text{param}} \cdot I_{\text{fast}}(n) + \mathbf{b}^f) \\ \tilde{y}^f(k+1) = \mathbf{W}_{\text{out}}^f \mathbf{u}^f(k+1) + b_{\text{out}}^f \end{cases} \quad (22)$$

where  $k$  denotes the time step and  $n$  is regarded as a constant value. Substituting the second expression of Equation 22 into the first one yields the autonomous dynamical system with the parameter  $I_{\text{fast}}(n)$  as follows:

$$\begin{aligned} \mathbf{u}^f(k+1) &= \tanh(\mathbf{W}^f \mathbf{u}^f(k) + \mathbf{W}_{\text{in}}^f \cdot (\mathbf{W}_{\text{out}}^f \mathbf{u}^f(k) + b_{\text{out}}^f) \\ &+ \mathbf{W}_{\text{param}} \cdot I_{\text{fast}}(n) + \mathbf{b}^f). \end{aligned} \quad (23)$$

The calculation of the LLE follows the standard approach using continuous Gram-Schmidt orthonormalization of the fundamental solutions to the linearized differential equation along the trajectory (Shimada and Nagashima, 1979), which is given by:

$$\delta \mathbf{u}(k+1) = J \cdot \delta \mathbf{u}(k), \quad (24)$$

where  $J$  is the Jacobian matrix of Equation 23, expressed as:

$$J = \frac{\partial \mathbf{u}^f(k+1)}{\partial \mathbf{u}^f(k)}. \quad (25)$$

Let  $r(k)$  be the argument of the hyperbolic tangent function in the RHS of Equation 23:

$$\begin{aligned} r(k) = & \mathbf{W}^f \mathbf{u}^f(k) + \mathbf{W}_{\text{in}}^f \cdot (\mathbf{W}_{\text{out}}^f \mathbf{u}^f(k) + b_{\text{out}}^f) \\ & + \mathbf{W}_{\text{param}} \cdot I_{\text{fast}}(n) + \mathbf{b}^f. \end{aligned} \quad (26)$$

Then, the Jacobian matrix can be described as follows:

$$\frac{\partial \mathbf{u}^f(k+1)}{\partial \mathbf{u}^f(k)} = \left( \mathbf{E} - \text{diag} \left[ \tanh^2(r(k)) \right] \right) \cdot \left( \mathbf{W}^f + \mathbf{W}_{\text{in}}^f \cdot \mathbf{W}_{\text{out}}^f \right). \quad (27)$$

## 2.7 Data generation models and observation

We generated time series to train the reservoir model using a nonlinear differential equation whose solutions were computed by numerical integration and discretized at specific time intervals  $\Delta t$ , with characteristic values for each model.

### 2.7.1 Lorenz equation

As an example of the function  $f_x$  described in the problem setting, we used the Lorenz 63 model to define a non-stationary signal source:

$$\begin{cases} \frac{dx_1}{dt} = a(x_2 - x_1) \\ \frac{dx_2}{dt} = -x_2 + x_1(\lambda - x_3) \\ \frac{dx_3}{dt} = -bx_3 + x_1x_2 \end{cases} \quad (28)$$

where  $a$  and  $b$  are parameters, and  $\lambda$  is considered to change slowly over time. The observation time series  $y(n)$  is given by:

$$y(n) = x_1(\Delta t \cdot n) \quad (29)$$

where  $n \in \mathbb{N}$  is the index of the discretized time steps. We set the parameter values to  $a = 10$  and  $b = 8/3$ , which are commonly employed, and discretized the time series with a time step of  $\Delta t = 0.05$ .

### 2.7.2 Rössler equation

The Rössler equation was also utilized as a data generation model:

$$\begin{cases} \frac{dx_1}{dt} = -x_2 - x_3 \\ \frac{dx_2}{dt} = x_1 + ax_2 \\ \frac{dx_3}{dt} = \lambda + x_3(x_1 - c) \end{cases} \quad (30)$$

where  $a$  and  $c$  are static parameters, and  $\lambda$  is considered to change slowly over time. The observation time series is  $y(n) = x_1(\Delta t \cdot n)$  as in the case of the Lorenz system, and the parameter values are set to  $a = 0.2, c = 5.7$ , and  $\Delta t = 0.7$ .

## 3 Results

### 3.1 Experiment 1: extraction of slow features from time series using the slow reservoir

Initially, we investigated the feasibility of observing parameter dynamics within a reservoir by feeding observed time series data from a nonlinear system with slowly changing parameter values into a reservoir characterized by a slow time constant. A schematic overview of the numerical computations performed in this study is depicted in Figure 1. The response of the slow reservoir to time series generated from the Lorenz system is illustrated in Figure 3, with a more detailed view provided in Figure 4 using a shorter time scale. In the Lorenz system described by Equation 28, the parameter  $\lambda$  varies slowly over time, following a triangular wave pattern between  $\lambda = 64$  and  $\lambda = 100$  (Figure 3A). Notably, the period of change in the parameter  $\lambda$  is 10,000 steps, which is two orders of magnitude larger than the typical timescale of the Lorenz attractor ( $\approx t = 1$ , which is equivalent to 20 steps with  $\Delta t = 0.05$ ). The time series of the variable  $x_1$  reflects these variations in parameter values, as shown in Figures 3B, C. Figure 4 presents the same data as Figure 3 but with a modified horizontal axis scale. At approximately  $t = 10,000$ , an abrupt change in the parameter value leads to a significant alteration in the shape of the  $x_1$  time series, as indicated in Figures 4B, C. Upon examining the internal state of the slow reservoir, we observed that certain nodes exhibited rapid temporal fluctuations (Figures 3D, 4D), while others displayed slower activities characterized by minimal high-frequency components (Figures 3E, 4E).

Our objective is to extract patterns of parameter fluctuations from the internal state of the slow reservoir in an unsupervised manner, assuming that the parameter's fluctuation is slower than the typical timescale of the time series. To accomplish this, we identified nodes exhibiting slow changes (see Materials and Methods). The nodes with the ten highest SD values are depicted in Figures 3D, 4D, while those with the five lowest values are shown in Figures 3E, 4E. Furthermore, we selected the 10% of nodes, i.e., 50 nodes, with the smallest SD values and calculated the negative mean of their absolute values, as illustrated in Figures 3F, 4F (see Materials and methods for details). As shown in Figure 3F,

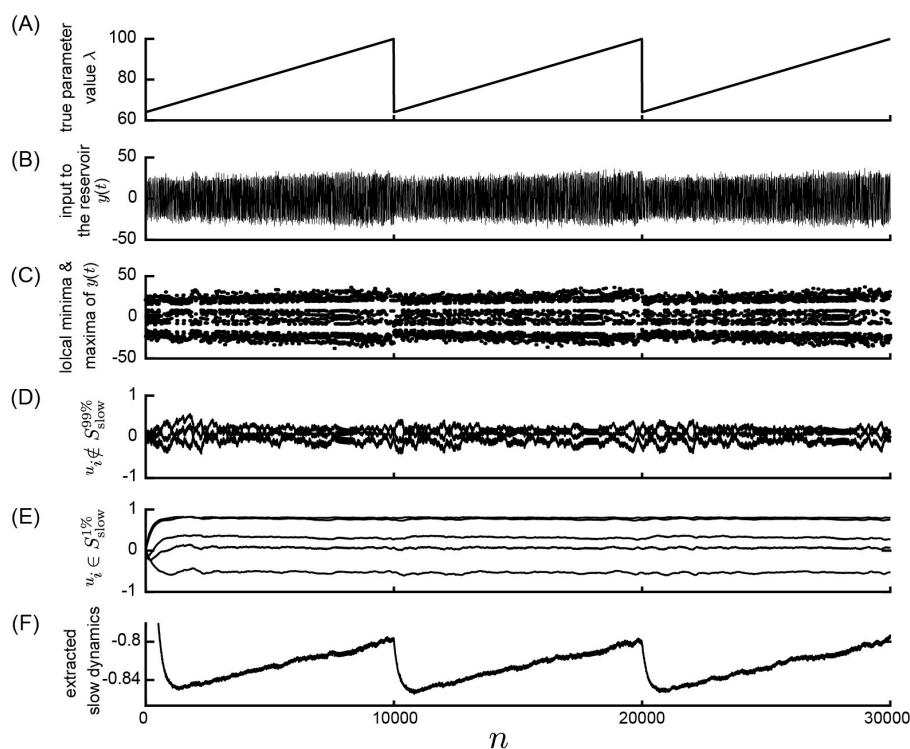


FIGURE 3

Response of the slow reservoir to time series generated by the Lorenz system in the Experiment 1. (A) True parameter value  $\lambda$  of the Lorenz system slowly changing from  $\lambda = 64$  to  $\lambda = 100$ . (B) Variable  $y(n) = x_1(\Delta t \cdot n)$ , representing the first element of the state of the Lorenz system used as the input to the reservoir. (C) Local minima and maxima of the trace shown in (B). (D, E) Values of the internal states,  $x_i$ , of the slow reservoir characterized by rapid and slow temporal fluctuations, respectively. (F) Extracted slow dynamics calculated as the average of the absolute values of internal nodes exhibiting slow behavior. (A–F) are plotted against time in the horizontal axis.

the average activity of the extracted slow nodes follows a trend similar to the temporal variation of the parameter  $\lambda$ . Given that approximately half of the nodes exhibit an inverted pattern, we took the absolute values of each node's values before averaging to ensure consistent directionality. The negative value of the final values is presented for easier comparison with the parameter  $\lambda$ 's fluctuation pattern. It's important to note that this process of taking the negative value, which relies on knowledge of the true parameter value, is unnecessary for predicting the unknown bifurcation presented in Subsection 3.2.

Figures 5, 6 show results of a similar analysis using the time series generated from the Rössler equation as input. Although the extracted slow dynamics in the Rössler attractor do not distinctly exhibit parameter variations as in the Lorenz attractor, the shape of the parameter variations remains observable, as demonstrated in Figures 5F, 6F.

Figure 7 illustrates the effect of varying the leak rate,  $\alpha$ , on the performance of the slow reservoir in extracting slow dynamics. Figure 7A shows the true time-varying parameter  $\lambda$ , which is used as a reference for comparison. Figures 7B–D show the extracted slow dynamics  $\tilde{u}^s(n)$  for three different leak rates:  $\alpha = 0.5$ ,  $\alpha = 0.99$ , and  $\alpha = 0.999$ , respectively. When the value of  $\alpha$  is too small compared to 1, the time constant of the slow reservoir becomes shorter than the time scale of the fluctuation of the parameter to be extracted, and it fails to reflect the true variation of the

parameter Figure 7B. On the other hand, when  $\alpha = 0.9999$  (i.e., when  $-\log_{10}(1 - \alpha) = -\log_{10}(0.0001) = 4$ ), the time constant of the reservoir becomes much longer than the time scale of the fluctuation of the parameter, as indicated by the relaxation behavior of the extracted slow dynamics (Figure 7D). Among Figures 7B–D, the correlation with the original time series of  $\lambda$  is highest when  $\alpha = 0.99$  as shown in Figure 7C. This suggests that there is an optimal value for  $\alpha$  in terms of the correlation. Figure 7E examines the correlation between the parameter  $\lambda$  and the extracted slow dynamics as the value of  $\alpha$  is varied, demonstrating the existence of this optimal value.

Within the framework of reservoir computing, regression to the training data from the internal states of the reservoir is a common practice. Based on the obtained results, it is clear that supervised learning regression can be applied to the time series of the parameter  $\lambda$  from the internal state of the slow reservoir (the results of supervised fitting are shown in Supplementary Figure S2, where very small RMSE is established). However, even without such supervised learning, if one can assume foresightedly that “the parameter variations have a much slower timescale than the typical timescale of the observed time series, allowing for the separation of timescales,” then, as demonstrated in this study, it might be possible to estimate the pattern of parameter variations simply by observing the activity of slowly moving nodes within the reservoir's internal state.



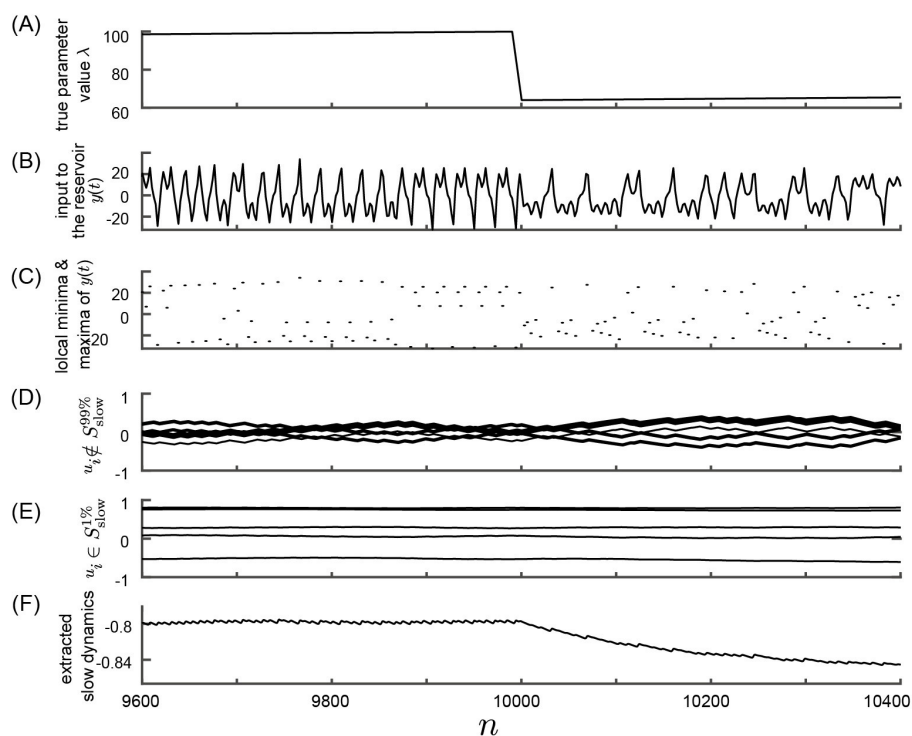


FIGURE 4

Response of the slow reservoir to time series generated by the Lorenz system in the Experiment 1 with an expanded time axis. This figure shows the same data as in Figure 3 but with an expanded time scale. (A) True parameter value  $\lambda$  of the Lorenz system. (B) Variable  $y(n) = x_1(\Delta t \cdot n)$ , representing the first element of the state of the Lorenz system used as the input to the reservoir. (C) Local minima and maxima of the trace shown in (B). (D, E) Values of the internal states,  $x_i$ , of the slow reservoir characterized by rapid and slow temporal fluctuations, respectively. (F) Extracted slow dynamics calculated as the average of the absolute values of internal nodes exhibiting slow behavior. (A–F) are plotted against time in the horizontal axis.

### 3.2 Experiment 2: prediction of unobserved bifurcation from time series data

As demonstrated in the previous subsection, extracting time series of slowly-varying elements from the slow reservoir allows us to unsupervisedly reveal the underlying slow parameter dynamics of the target system. As discussed in the Introduction, Patel et al. (2021) and Kim et al. (2021) have shown that by concurrently inputting the time series of actual parameter values into the reservoir, it is possible to predict bifurcations in the system's attractors, even if these bifurcations are not present in the training data. In this case, we present a scenario where the slow dynamics, unsupervisedly extracted by the slow reservoir, are fed into the reservoir separately from the observations of the fast dynamics, enabling the prediction of bifurcations in the target system that are not contained in the training data. Our model, depicted in Figure 2, consists of three reservoirs: the slow reservoir, the slow dynamics predictor, and the fast reservoir. The slow reservoir is characterized by long time constants in its leaky units and a spectral radius equal to 1, specifically engineered to extract the slowest-moving dynamics by calculating the mean absolute values of the 10% most slowly changing elements. The slow dynamics predictor receives outputs from the slow reservoir and learns its dynamics. The output from this slow reservoir is smoothed through a linear filter before being sent to the two downstream reservoirs (Methods). Meanwhile, the fast reservoir receives both the fast dynamics directly observed

from the target system and the slow dynamics extracted by the slow reservoir, predicting the subsequent state of  $y(n)$ . We use time series data generated from the Lorenz attractor with a slowly varying parameter as the target system to learn. Figure 8 illustrates the results of learning by the model.

Figure 8 illustrates the results of the model's learning process. In this numerical experiment, the parameter  $\lambda$  changes linearly and gradually over time (Figure 8) under a scenario where bifurcations occur in the fast dynamics leading to the disappearance of the chaotic attractor (Figure 8). Beyond  $n = 7,500$ , the Lorenz system develops a stable fixed point characterized by two complex conjugate eigenvalues. As shown in Figure 8B, chaotic oscillations occur prior to  $n = 7,500$ , but suddenly cease at a specific point. After  $n = 7500$ , the system converges toward one of the two stable fixed points. Figure 8C depicts the value of the LLE for the Lorenz system when the parameter value displayed in Figure 8A remains constant over time. Note that this LLE is computed for each point along the horizontal axis with a fixed value of  $\lambda$ , unlike in a system with a temporally varying  $\lambda$ . As shown in Figure 2A, the model operates as an open-loop model from  $n = 0$  to  $n = 5500$ , driven by the external input  $y(n)$  depicted in Figure 8B. The slow dynamics extracted from the reservoir after smoothing by the linear filter are presented in Figure 8D. The model undergoes a transient phase up to  $n = 1,000$ , but then exhibits nearly linear behavior resembling the true parameter variations shown in Figure 8A. Starting at  $n =$

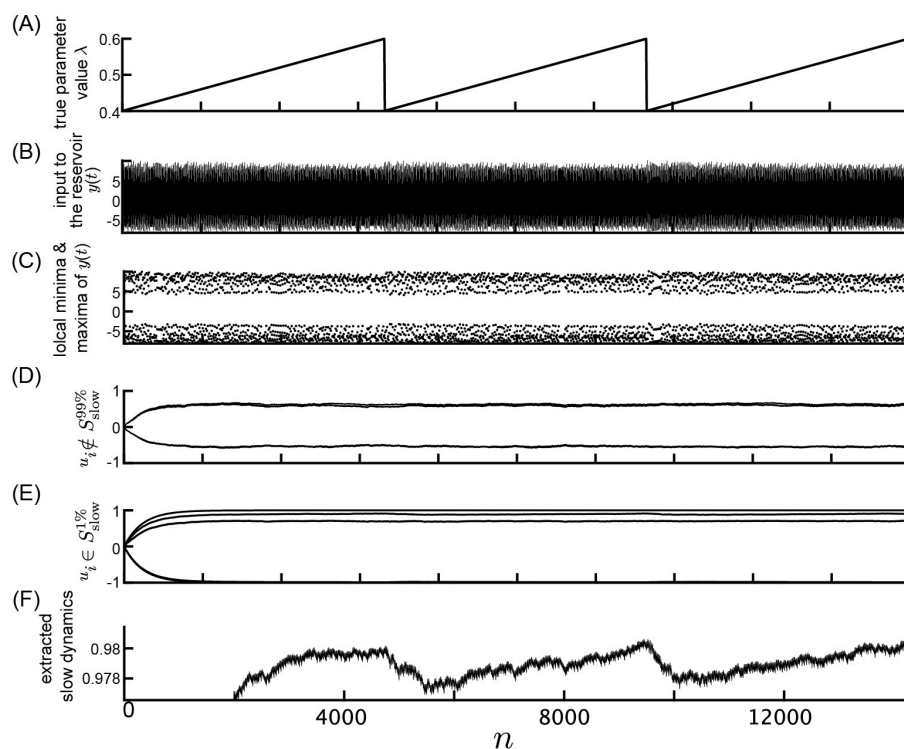


FIGURE 5

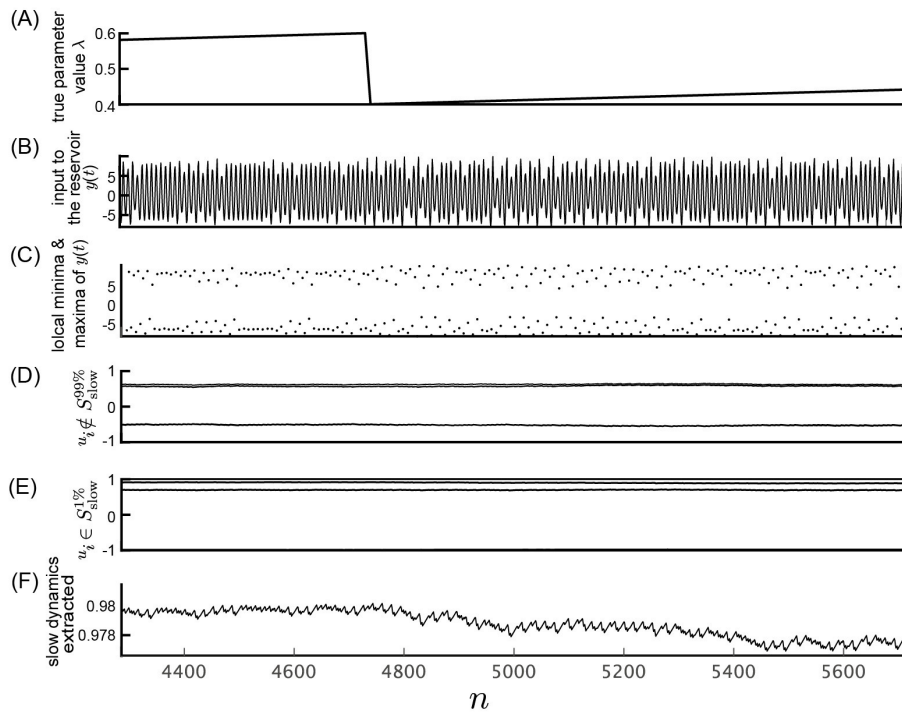
Response of the slow reservoir to time series generated by the Rössler equation. (A) True parameter value  $\lambda$  of the Rössler equation. (B) Variable  $y(n) = x_1(\Delta t \cdot n)$ , representing the first element of the state of the Rössler equation used as the input to the reservoir. (C) Local minima and maxima of the trace shown in (B). (D, E) Values of the internal states,  $x_i$ , of the slow reservoir characterized by rapid and slow temporal fluctuations, respectively. (F) Extracted slow dynamics calculated as the average of the absolute values of internal nodes exhibiting slow behavior. (A–F) are plotted against time in the horizontal axis.

5,500, the slow dynamics predictor is changed to a closed-loop model, generating  $\tilde{h}(n)$ , the prediction of  $h(n)$ . At this point, the internal state of the reservoir, determined by the external force during the training phase, remains unchanged; only the input is instantaneously replaced by the feedback from its own output. Figure 8E shows the output from the fast reservoir. The results of fitting  $y(n)$  (the time series shown in Figure 8B) from  $n = 1,500$  to  $n = 5,500$  are marked in red. After  $n = 5,500$ ,  $\tilde{h}(n)$  generated by the slow dynamics predictor, along with the feedback from its own output, are fed to the fast reservoir to generate predictions for  $y(n)$ . At this stage, the entire system operates as an autonomous system with no external input. By  $n = 7,000$ , chaotic oscillations disappear, and the trajectory converges to a stable fixed point, as shown in Figure 8E.

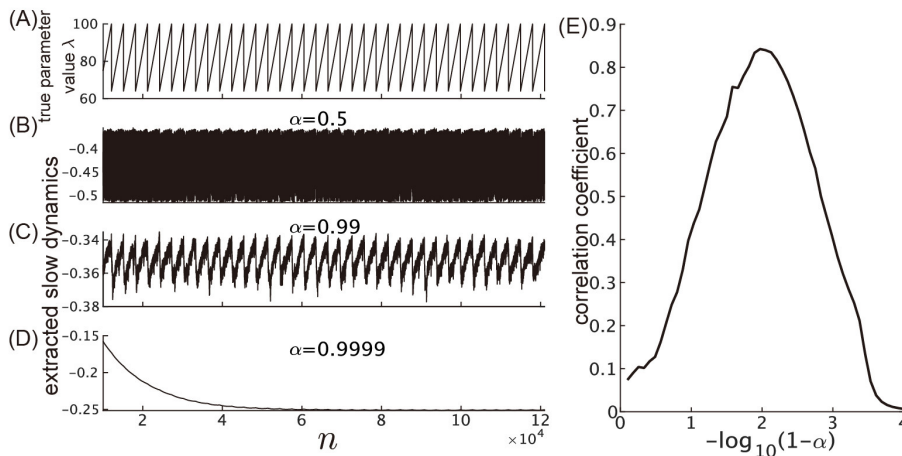
For each time point along the horizontal axis in Figure 8E, the LLE of the closed-loop fast reservoir is estimated by fixing the value of the input  $h(n)$  to that at each time point in Figure 8D. It was found that, during the chaotic oscillations observed in the fast reservoir's output before  $n = 6,000$ , the LLE closely matches that of the Lorenz system during the learning phase (Figure 8F). However, the system still does not fully replicate the slight decreasing tendency of the LLE and the narrow windows with zero LLE found in the original Lorenz system. After  $n = 7,000$ , the LLE of the fast reservoir takes negative values. Using the fixed values of the input to the fast reservoir,  $\tilde{h}(n = 6,000)$ , the attractor reconstructed by the delay time coordinate shows a

shape akin to that of the Lorenz attractor (Figure 8G). Furthermore, when using the values of  $\tilde{h}(n)$  after the oscillations have ceased, the trajectory converges with rotation around a single stable fixed point (Figures 8H, I). This suggests that the existence in the original Lorenz system of a stable fixed point with one real eigenvalue and two conjugate complex eigenvalues is predicted only by learning from the chaotic time series as  $\lambda$  decreases. Overall, the results shown in Figure 8 suggest that, solely by observing the fast variable, the model successfully learned the dynamical flow of the original Lorenz system, including the shape of the attractor, its stability, and its slowly changing vector field.

Figure 9 shows the model's behavior with different values of the training data length. We assess the model's ability to predict bifurcations in the Lorenz system by varying the lengths of training data. The experimental setting is the same as in the Figure 8, except that the lengths of training data are varied. Figure 8A shows the true parameter  $\lambda$  decreasing linearly over time, while Figures 8B–E display the model's prediction performance with training data of different lengths. In Figure 8B, the model uses 4,000 steps of training data, which is the same setting as in the main text. In Figure 8C, we reduce the length of the training data to 2,500 steps. Despite this shorter training phase, the model still captures the qualitative property of the bifurcation dynamics where it converges to a stable fixed point with complex eigenvalues after the bifurcation occurs. In Figure 8D, the length of the training data is 1,200 steps, and the model still predicts the bifurcation in the



**FIGURE 6** Response of the slow reservoir to time series generated by the Rössler equation with an expanded time axis. This figure shows the same data as in fig. 5 but with an expanded time scale. **(A)** True parameter value  $\lambda$  of the Rössler equation. **(B)** Variable  $y(n) = x_1(\Delta t \cdot n)$ , representing the first element of the state of the Rössler equation used as the input to the reservoir. **(C)** Local minima and maxima of the trace shown in **(B)**. **(D, E)** Values of the internal states,  $x_i$ , of the slow reservoir characterized by rapid and slow temporal fluctuations, respectively. **(F)** Extracted slow dynamics calculated as the average of the absolute values of internal nodes exhibiting slow behavior. **(A–F)** are plotted against time in the horizontal axis.



**FIGURE 7** Dependence of the slow reservoir’s performance on the leak rate. **(A)** Time course of the true parameter value  $\lambda$ . **(B–D)** The time courses of the extracted slow dynamics,  $\tilde{u}^s(n)$ , for  $\alpha = 0.5, 0.99, 0.9999$ , respectively. **(E)** The Pearson’s correlation coefficient between  $\lambda$  and  $\tilde{u}^s(n)$ .

predicting phase. Finally, in [Figure 8E](#), where only 1,000 steps of training data are used, the model fails to predict the bifurcation. The time series output shows that the system continues to remain in a chaotic regime without transitioning to the stable fixed point observed in the Lorenz system. These results highlight the

importance of training data length in capturing the full dynamical behavior of the system, particularly in predicting bifurcations. While the model can anticipate the bifurcation with relatively short training data, its precision and accuracy improve with longer training phases.

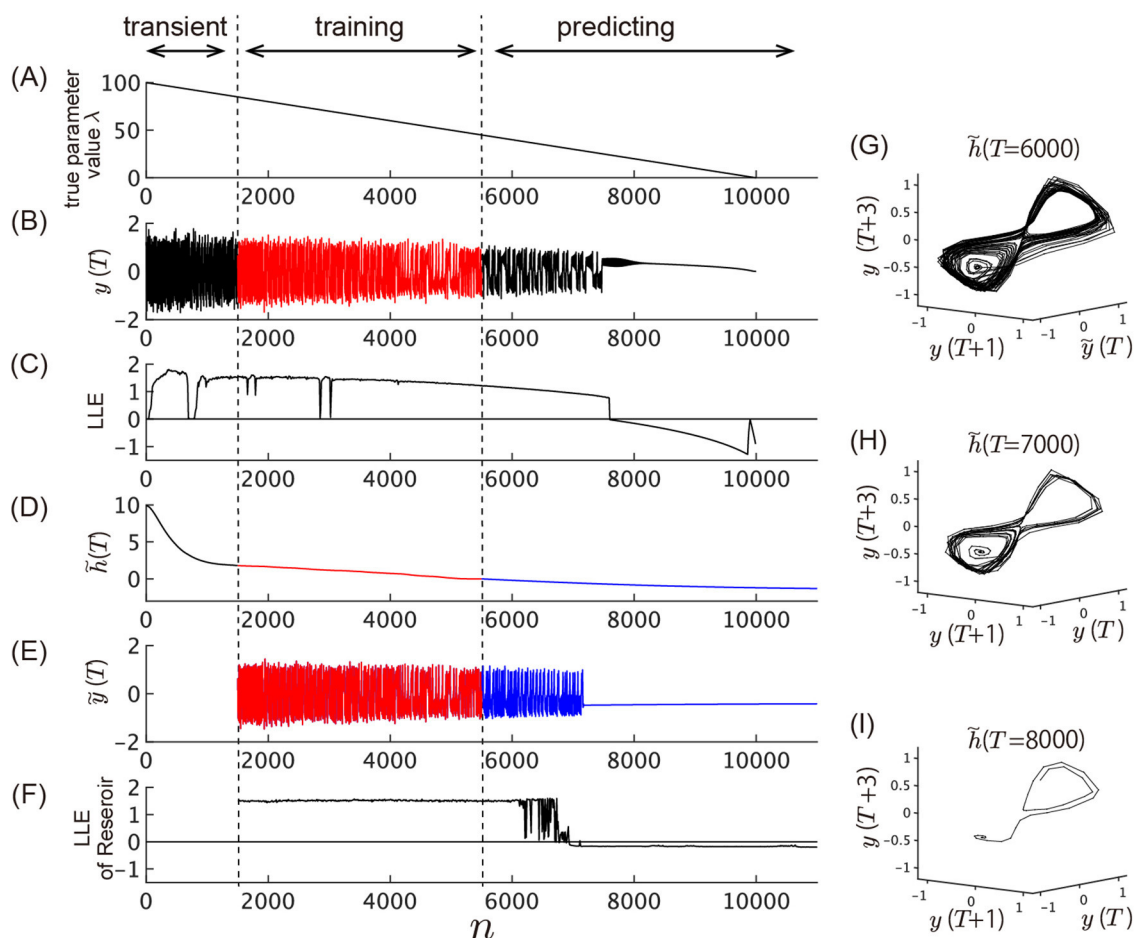


FIGURE 8

Prediction of unobserved bifurcation. This figure presents the time series during consecutive training and predicting phases. (A–C) Time evolution of the Lorenz system that generates the data to be learned by the model. (A) Slowly varying true parameter value  $\lambda$  of the Lorenz system plotted against time. (B) Time series generated by the Lorenz system,  $y(n) = x_1(\Delta t \cdot n)$ , where  $x_1(t)$  is the first variable of the Lorenz system. The time series from  $n = 1,500$  to  $n = 5,500$  is used as training data and fed into the model. Bifurcation occurs around  $n = 7,500$ , where the chaotic oscillation vanishes. After this point, the trajectory converges to a stable fixed point with one real eigenvalue and two complex conjugate eigenvalues. (C) LLE of the Lorenz system corresponding to the value of  $\lambda$  shown in (A). The LLE is plotted against time along the horizontal axis, with each value calculated with a static value of  $\lambda$  corresponding to the same time point in (A). (D–F) Model outputs. From  $n = 0$  to  $n = 5,500$ , the model is driven by the external input  $y(n)$  shown in (B). After  $n = 5,500$ , the model switches to the closed-loop model depicted in Figure 2. (D) Slow dynamics extracted by the slow reservoir ( $n = 0$  to  $n = 5,500$ ) and prediction of its dynamics by the slow dynamics predictor after  $n = 5,500$  (blue line). The extracted slow dynamics before  $n = 1,500$  are treated as transient dynamics and not used as training data. (E) Output of the fast reservoir. The red line plotted between  $n = 1,500$  and  $n = 5,500$  shows the fitting of the training data shown in (B). The blue line plotted after  $n = 5,500$  shows the prediction of the data shown in (B) by the closed-loop model, showing the bifurcation from chaotic oscillation to oscillation death. (F) LLE of the closed-loop fast reservoir, calculated with a fixed value of the external input  $p$ . The value of  $p$  is set to the value of  $h(n)$  or  $\tilde{h}(n)$  plotted at the same time point in (D). (G–I) Reconstructed attractor shape by delay embedding of the fast reservoir output  $y(n)$  with the input to the fast reservoir at  $\tilde{h}(n = 6,000)$ ,  $\tilde{h}(7,000)$ , and  $\tilde{h}(8,000)$ . The trajectories converge to a stable fixed point in (H, I).

Lastly, we comment on the computational cost of this model. Using an Apple MacBook Pro 2021 with an M1 Max processor and 32GB of memory, along with MATLAB, the computation time for the training phase in the numerical simulations shown in Figure 8 takes approximately 1 second or less for the slow reservoir, slow reservoir predictor, and fast reservoir (all of which are 500-dimensional discrete-time mapping models). Additionally, the training of the reservoir outputs is performed using Ridge regression, as shown by the analytical solution in Equation 17, with a computation time of less than 0.1 s.

## 4 Discussion

We have demonstrated that unsupervised extraction of the very slowly changing parameters of the dynamical system generating the signals is possible by simply feeding the observation to a reservoir with a long-time scale and selecting the internal nodes of the reservoir with slowly varying states. Furthermore, we have shown this reservoir's capability to predict bifurcations not present in the training data, such as the death of chaotic oscillations, by inputting the extracted slow features and observation signal into another reservoir. Kim et al. (2021) and Patel et al. (2021) demonstrated

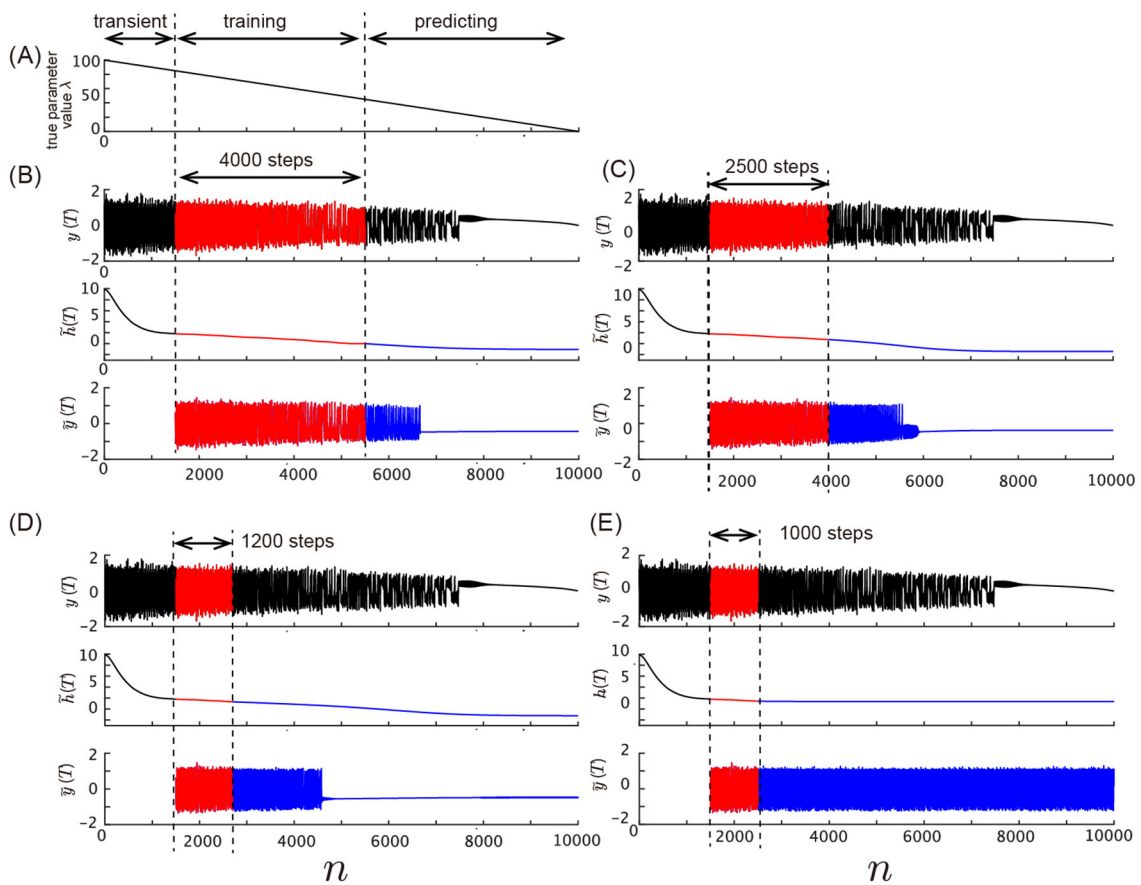


FIGURE 9

Effect of training data length on the prediction of bifurcation. (A) Slowly varying true parameter value  $\lambda$  plotted against time. (B–E) The training data and time series generated by the model using varying lengths of training data. (B) Time series with training data length of 4,000 steps, which captures the bifurcation occurring after  $n = 7,500$ . The red section shows the training phase, and the blue section shows the predicted time series. (C) Time series with training data length of 2,500 steps. Similar to (B), the model captures the bifurcation qualitatively. (D) Time series with training data length of 1,200 steps. The model still predicts the bifurcation, but the accuracy begins to degrade. (E) Time series with training data length of 1,000 steps, where the model fails to predict the bifurcation.

the prediction of unobserved bifurcations not present in the training data using a reservoir computing framework. Their work illustrated the remarkable capability of reservoir computing to learn the parameter dependencies within dynamical system flows and to reproduce unknown bifurcations. However, they treated the parameters as known, which is not the case in real-world applications, where the values of these parameters often cannot be observed. In this study, we introduce two reservoirs: a slow feature predictor that forecasts the movement of these slow features, and a fast reservoir that predicts the values of the observed time series. By inputting the slow features resulting from the unsupervised extraction, we establish a closed-loop model that operates as a fully autonomous dynamical system during the predicting phase. This demonstrates the ability to forecast the emergence of unknown bifurcations without any direct observation of the parameter value. Nonlinear, non-stationary processes are abundant in various natural and physical phenomena. Additionally, numerous scenarios probably exist where slow dynamics inducing qualitative dynamics changes remain unobservable. The potential

applications of this approach are vast, spanning fields such as neuroscience (including electrophysiological measurements, electroencephalography, functional Magnetic Resonance Imaging, and disease progression with tipping points), material science (including surface science), and weather prediction and control.

The main limitation of the current work is the lack of understanding of the mechanism underlying the phenomenon wherein the behavior of slowly moving nodes, selected heuristically from within the slow reservoir, is similar to variations in the original system's parameters. As mentioned in the Introduction, the observations made could be explained if the reservoir can achieve generalized synchronization with the target system (Carroll, 2020; Rulkov et al., 1995), including the slow parameter dynamics. For the results shown in Figure 3, we conducted the same numerical simulation using a reservoir with linear dynamics by replacing the activation function in Equation 8 with the identity map. The results show that a linear reservoir with a slow time constant does not yield parameter estimation, even with

supervised training, where the internal state of the reservoir is fitted to the true parameter value (Supplementary Figure S2). This suggests that the nonlinearity of the reservoir is crucial for the current results. However, generalized synchronization would not fully explain the current results. For example, fast-moving nodes also exist within the slow reservoir. It is not trivial that in the internal state of the reservoir,  $\mathbf{u}^s \in \mathbb{R}^{N^s}$ , the directions of fast and slow fluctuations align along axes,  $u_1, u_2, \dots, u_{N^s}$  (i.e., different nodes). In fact, it would not be surprising if fast and slow fluctuations were superimposed at all single nodes. Therefore, further investigation is required to elucidate the logical reason why simply selecting slow-moving nodes worked well as a heuristic. Conversely, employing a more sophisticated method to separate the directions of fast and slow fluctuations might lead to better performance (Antonelo and Schrauwen, 2012).

Previous works have extensively explored the behavior of complex systems around tipping points (Dakos et al., 2008; Veraart et al., 2011; Liu et al., 2013). For instance, the Dynamical Network Biomarker (DNB) method captures the increase in temporal fluctuations and the intensified correlation associated with critical slowing down Liu et al. (2013). Unlike the approach in the current study, which involves learning the flow of the dynamical system in a relatively low-dimensional, deterministic, and strongly nonlinear phase space, the DNB method utilizes the generic behavior near bifurcation points in very high-dimensional systems based on linearization around a fixed point. Given their distinct advantages, combining these methods in the future might improve the prediction and control of non-stationary, nonlinear systems.

Kim et al. (2021) the emergence of chaotic attractors in the Lorenz system by extrapolating the parameter space and learning in regions without chaotic attractors, where only two stable fixed points exist. In our research, we have extracted the slowly changing parameters of the target system by receiving its generated time series through the reservoir. However, applying our method to predict the emergence of a chaotic strange attractor by learning observation from the Lorenz system with stable fixed points is currently challenging because our method relies on observing long time series to extract slow features, whereas the target system does not produce a long time series with oscillations if the trajectory converges to a fixed point. A new framework would be necessary to estimate the parameter changes in a system with stable fixed points, e.g., by introducing external perturbations to the target system and receiving its response through the reservoir.

## Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

## References

Antonelo, E., and Schrauwen, B. (2012). Learning slow features with reservoir computing for biologically-inspired robot localization. *Neural Netw.* 25, 178–190. doi: 10.1016/j.neunet.2011.08.004

## Author contributions

KT: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. YK: Conceptualization, Funding acquisition, Methodology, Writing – review & editing.

## Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by JSPS KAKENHI (Nos. 20K19882, 20H04258, 20H00596, 21H05163, 23K11259, 23H03468, and 24H02330) and the Japan Science and Technology Agency (JST) Moonshot R&D (JPMJMS2284 and JPMJMS2389), JST CREST (JPMJCR17A4), JST ALCA-Next (JPMJAN23F3). This paper is also based on results obtained from a project JPNP16007 commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

## Acknowledgments

The authors acknowledge the use of ChatGPT-4o, a large language model developed by OpenAI, for language editing and assistance in improving the clarity of the manuscript. The tool was used for language correction purposes only, and all intellectual content remains the responsibility of the authors.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/frai.2024.1451926/full#supplementary-material>

Bnard, H. (1901). Les tourbillons cellulaires dans une nappe liquide. mthodes optiques dobservation et denregistrement. *Journal de Physique Thorique et Applique* 10, 254–266. doi: 10.1051/jphysap:0190100100025400

- Buzski, G. (2002). Theta oscillations in the hippocampus. *Neuron* 33, 325–340. doi: 10.1016/S0896-6273(02)00586-X
- Carroll, T. L. (2020). Dimension of reservoir computers. *Chaos: Interdisc. J. Nonlinear Sci.* 30:013102. doi: 10.1063/1.5128898
- Dakos, V., Scheffer, M., van Nes, E. H., Brovkin, V., Petoukhov, V., and Held, H. (2008). Slowing down as an early warning signal for abrupt climate change. *Proc. Nat. Acad. Sci.* 105, 14308–14312. doi: 10.1073/pnas.0802430105
- Ertl, G. (1991). Oscillatory kinetics and spatio-temporal self-organization in reactions at solid surfaces. *Science* 254, 1750–1755. doi: 10.1126/science.254.5039.1750
- Gallicchio, C., Micheli, A., and Pedrelli, L. (2017). Deep reservoir computing: A critical experimental analysis. *Neurocomputing* 268, 87–99. doi: 10.1016/j.neucom.2016.12.089
- Itoh, H., and Kimoto, M. (1996). Multiple attractors and chaotic itinerancy in a quasigeostrophic model with realistic topography: implications for weather regimes and low-frequency variability. *J. Atmosph. Sci.* 53, 2217–2231. doi: 10.1175/1520-0469(1996)053<2217:MAACII>2.0.CO;2
- Itoh, Y. (2023). Verifying the robustness of using parameter space estimation with ridge regression to predict a critical transition. *Nonlinear Theory Appl. IEICE* 14, 579–589. doi: 10.1587/nolta.14.579
- Jaeger, H. (2008). *Discovering Multiscale Dynamical Features with Hierarchical Echo State Networks*. Jacobs University Technical Report, No. 10, p. 1–30.
- Jaeger, H., and Haas, H. (2004). Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* 304, 78–80. doi: 10.1126/science.1091277
- Jaeger, H., Lukoevius, M., Popovici, D., and Siewert, U. (2007). Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Netw.* 20, 335–352. doi: 10.1016/j.neunet.2007.04.016
- Kaneko, K., and Tsuda, I. (2003). Chaotic itinerancy. *Chaos: Interdisc. J. Nonlinear Sci.* 13, 926–936. doi: 10.1063/1.1607783
- Katori, Y. (2019). “Hierarchical network model of predictive coding based on reservoir computing,” in *The 2019 International Symposium on Nonlinear Theory and Its Applications (NOLTA2019)* (Institute of Electronics, Information and Communication Engineers), 341–344.
- Katori, Y., Sakamoto, K., Saito, N., Tanji, J., Mushiaki, H., and Aihara, K. (2011). Representational switching by dynamical reorganization of attractor structure in a network model of the prefrontal cortex. *PLoS Comp. Biol.* 7:e1002266. doi: 10.1371/journal.pcbi.1002266
- Kim, J., Lu, Z., Nozari, E., Pappas, G., Bassett, D. S. (2021). Teaching recurrent neural networks to infer global temporal structure from local examples. *Nat. Mach. Intell.* 3, 316–323. doi: 10.1038/s42256-021-00321-2
- Kong, L.-W., Fan, H.-W., Grebogi, C., and Lai, Y.-C. (2021). Machine learning prediction of critical transition and system collapse. *Phys. Rev. Res.* 3:013090. doi: 10.1103/PhysRevResearch.3.013090
- Liu, R., Aihara, K., and Chen, L. (2013). Dynamical network biomarkers for identifying critical transitions and their driving networks of biologic processes. *Quant. Biol.* 1:105–114. doi: 10.1007/s40484-013-0008-0
- Lu, Z., and Bassett, D. S. (2020). Invertible generalized synchronization: A putative mechanism for implicit learning in neural systems. *Chaos* 30:063133. doi: 10.1063/5.0004344
- Lu, Z., Hunt, B. R., and Ott, E. (2018). Attractor reconstruction by machine learning. *Chaos* 28:061104. doi: 10.1063/1.5039508
- Lu, Z., Pathak, J., Hunt, B., Girvan, M., Brockett, R., and Ott, E. (2017). Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. *Chaos* 27:041102. doi: 10.1063/1.4979665
- Manneschi, L., Ellis, M. O. A., Gigante, G., Lin, A. C., Del Giudice, P., and Vasilaki, E. (2021). Exploiting multiple timescales in hierarchical echo state networks. *Front. Appl. Mathemat. Statist.* 6:616658. doi: 10.3389/fams.2020.616658
- Marwan, N., and Kraemer, K. H. (2023). Trends in recurrence analysis of dynamical systems. *Eur. Phys. J. Spec. Topics* 232, 5–27. doi: 10.1140/epjs/s11734-022-00739-8
- Patel, D., Canaday, D., Girvan, M., Pomerance, A., and Ott, E. (2021). Using machine learning to predict statistical properties of non-stationary dynamical processes: System climate, regime transitions, and the effect of stochasticity. *Chaos* 31:033149. doi: 10.1063/5.0042598
- Raab, M., Zeininger, J., Suchorski, Y., Tokuda, K., and Ruppelrechter, G. (2023). Emergence of chaos in a compartmentalized catalytic reaction nanosystem. *Nat. Commun.* 14:1. doi: 10.1038/s41467-023-36434-y
- Rabinovich, M., Volkovskii, A., Lecanda, P., Huerta, R., Abarbanel, H. D. I., and Laurent, G. (2001). Dynamical encoding by networks of competing neuron groups: winnerless competition. *Phys. Rev. Lett.* 87:068102. doi: 10.1103/PhysRevLett.87.068102
- Rulkov, N. F., Sushchik, M. M., Tsimring, L. S., and Abarbanel, H. D. I. (1995). Generalized synchronization of chaos in directionally coupled chaotic systems. *Phys. Rev. E* 51, 980–994. doi: 10.1103/PhysRevE.51.980
- Shimada, I., and Nagashima, T. (1979). A numerical approach to ergodic problem of dissipative dynamical systems. *Prog. Theoret. Phys.* 61, 1605–1616. doi: 10.1143/PTP.61.1605
- Steriade, M., Nunez, A., and Amzica, F. (1993). A novel slow (< 1 Hz) oscillation of neocortical neurons in vivo: depolarizing and hyperpolarizing components. *J. Neurosci.* 13, 3252–3265. doi: 10.1523/JNEUROSCI.13-08-03252.1993
- Tamura, H., Katori, Y., and Aihara, K. (2019). Possible mechanism of internal visual perception: Context-dependent processing by predictive coding and reservoir computing network. *J. Robot. Network. Artif. Life* 6, 42–47. doi: 10.2991/jrnal.k.190531.009
- Tanaka, G., Matsumori, T., Yoshida, H., and Aihara, K. (2022). Reservoir computing with diverse timescales for prediction of multiscale dynamics. *Phys. Rev. Res.* 4:L032014. doi: 10.1103/PhysRevResearch.4.L032014
- Tokuda, K., Katori, Y., and Aihara, K. (2019). Chaotic dynamics as a mechanism of rapid transition of hippocampal local field activity between theta and non-theta states. *Chaos* 29:113115. doi: 10.1063/1.5110327
- Veraart, A. J., Faassen, E. J., Dakos, V., van Nes, E. H., Lürling, M., and Scheffer, M. (2011). Recovery rates reflect distance to a tipping point in a living system. *Nature* 481, 357–359. doi: 10.1038/nature10723
- Vohryzek, J., Deco, G., Cessac, B., Kringelbach, M. L., and Cabral, J. (2020). Ghost attractors in spontaneous brain activity: recurrent excursions into functionally-relevant bold phase-locking states. *Front. Syst. Neurosci.* 14:20. doi: 10.3389/fnsys.2020.00020
- Waddington, C. (1961). “Genetic assimilation,” in *Advances in Genetics* (Cambridge, MA: Academic Press), 257–293.
- Wiskott, L., and Sejnowski, T. J. (2002). Slow feature analysis: unsupervised learning of invariances. *Neural Comp.* 14, 715–770. doi: 10.1162/089976602317318938
- Yonemura, Y., and Katori, Y. (2021). Network model of predictive coding based on reservoir computing for multi-modal processing of visual and auditory signals. *Nonlinear Theory Appl. IEICE* 12, 143–156. doi: 10.1587/nolta.12.143
- Zhai, Z.-M., Moradi, M., Glaz, B., Haile, M., and Lai, Y.-C. (2024). Machine-learning parameter tracking with partial state observation. *Phys. Rev. Res.* 6:013196. doi: 10.1103/PhysRevResearch.6.013196