Unsupervised learning for anticipating critical transitions

Shirin Panahi,[1] Ling-Wei Kong,[2] Bryan Glaz,[3] Mulugeta Haile,[4] and Ying-Cheng Lai[1,5,*]

[1] *School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287, USA*

[2] *Department of Computational Biology, Cornell University, Ithaca, NY 14850, USA*

[3] *Vehicle Technology Directorate, CCDC Army Research Laboratory, 2800 Powder Mill Road, Adelphi, MD 20783-1138, USA*

[4] *Vehicle Technology Directorate, CCDC Army Research Laboratory, 6340 Rodman Road, Aberdeen Proving Ground, MD 21005-5069, USA*

[5] *Department of Physics, Arizona State University, Tempe, Arizona 85287, USA*

(Dated: January 2, 2025)

For anticipating critical transitions in complex dynamical systems, the recent approach of parameter-driven reservoir computing requires explicit knowledge of the bifurcation parameter. We articulate a framework combining a variational autoencoder (VAE) and reservoir computing to address this challenge. In particular, the driving factor is detected from time series using the VAE in an unsupervised-learning fashion and the extracted information is then used as the parameter input to the reservoir computer for anticipating the critical transition. We demonstrate the power of the unsupervised learning scheme using prototypical dynamical systems including the spatiotemporal Kuramoto–Sivashinsky system. The scheme can also be extended to scenarios where the target system is driven by several independent parameters or with partial state observations.

In complex dynamical systems ranging from ecological [1] and climate systems [2] to infrastructure [3] and social [4, 5] networks, critical transitions can occur in which the system undergoes an abrupt and often catastrophic switch to a characteristically different final state [4, 6]. Anticipating critical transitions is important to disaster prediction and prevention, risk mitigation, and resilience enhancement [7, 8]. If an accurate model of the system is available with the knowledge of the bifurcation parameter and its possible variation into the future, computations can be carried out to anticipate a critical transition. The typical real-world scenario is that the mathematical model of the underlying system is not available, so one must rely on observational or measurement data to anticipate critical transitions but this can be quite challenging.

There are two main approaches to data-based anticipation of critical transitions [Sec. S1 in Supplementary Information (SI) [9]]. One is based on finding the system equations from data using sparse optimization [10], which is effective if the equation structure of the underlying system is particularly simple. The second approach is based on machine learning [11–17]. A recent method [13, 17] is parameter-driven reservoir computing [18, 19] for nonlinear and complex dynamical systems [20–25]. Specifically, in conventional reservoir computing, the neural network is trained to learn the "dynamical climate" of the target system. However, parameter-driven reservoir computing also enables the neural network to learn how the dynamical climate changes with a system parameter [13, 17], which was applied to anticipating crises [13], abrupt onset or destruction of synchronization in coupled oscillator systems [15], amplitude death [26], and the occurrence of periodical windows [27]. Parameter-driven reservoir computing has also been exploited for constructing digital twins of nonlinear systems [17] and for anticipating tipping in real-world systems such as the potential collapse of Atlantic Meridional Overturning Circulation [28]. A limitation is that the knowledge of the time-dependent bifurcation parameter is required.

In this Letter, we develop a machine-learning framework that combines a variational autoencoder (VAE) with parameter-driven reservoir computing to address the challenge of anticipating critical transitions without explicit knowledge of the bifurcation parameter. In particular, we use VAE to extract the parameter that drives the system towards a critical point from the available time series data. This is essentially *unsupervised* learning leading to a "parameter" of the system and its variations. With this information, parameter-driven reservoir computing is then exploited to anticipate a critical transition. We demonstrate the power of the unsupervised learning scheme using prototypical nonlinear dynamical systems. We also generalize the framework to scenarios where the target system is driven by more than one independent parameter or only partial state observation is available. While the previous parameter-driven reservoir-computing scheme is data-driven and requires no system models, it still needs knowledge about the bifurcation parameter whose variations leads to a critical transition. Our unsupervised-learning framework relaxes this requirement in that no prior knowledge of the system parameter is needed. In fact, the bifurcation parameter and its variations can be faithfully extracted from data. This makes machine-learning based prediction of critical transitions a step closer to real applications.

Figure 1 outlines the main features of our proposed architecture of unsupervised leaning for anticipating critical transitions, tailored to scenarios where the bifurcation parameter is not accessible. A VAE is first trained

* Ying-Cheng.Lai@asu.edu

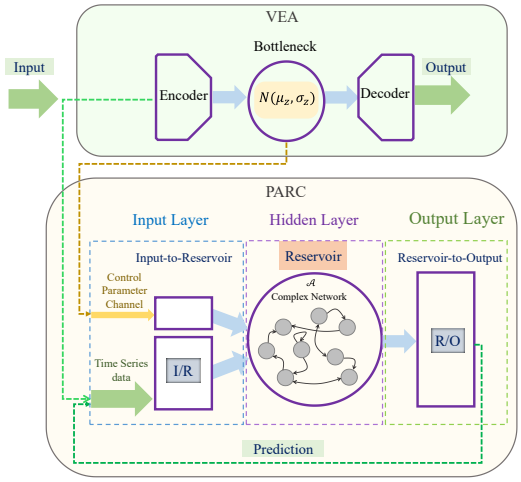

FIG. 1. Integrated architecture of VAE and parameter-driven reservoir computing. The VAE consists of the encoder and the decoder, with the latent distribution parameter extracted from the input data in the bottleneck, and it identifies the bifurcation parameter and extracts its variations. The input to the reservoir computer consist of the time-series data from the target system and the VAE-extracted parameter variations, and the output is the prediction of long-term dynamics.

to extract the parameters characterizing the system dynamics and provide multi-step prediction. In particular, the VAE has an encoder and a decoder. When presented with a time-series dataset, the encoder extracts the latent parameters generating the dynamic variations in the dataset, and the decoder employs the latent parameters encapsulating essential information about the dynamics to reconstruct the time series from the given initial condition. The trained VAE serves as the input parameter component in parameter-driven reservoir computing.

Since the purpose of the VAE is processing time series to extract information about the parameter variations, we use a deep convolution neural network (DCNN) as the encoder. The DCNN detects dynamical features from the input time series and outputs μ and σ that parameterize the Gaussian distribution of the bifurcation parameter z for each input x. For the decoder, we employ a feedforward neural network (FNN) that propagates a given initial condition x_0 for several time steps through a feedback loop. Multiple-step predictions $\hat{x}$ are generated via iterative single-step prediction by the decoder FNN, where the output of the last one-step prediction $\hat{x}$ becomes the input for the next iteration. All iterations are also modulated by the extracted parameter z. That is, the decoder works in a similar fashion to parameter-driven reservoir computing, but with a simpler structure that allows backpropagation through the entire VAE. The initial value x_0 is chosen from a random training sample x. The training goal of the VAE is reconstructing the time series immediately after x_0 for a certain length. Overall, during the training phase, the VAE is optimized to align the output of the decoder with a time-series example from the normal region of the dataset. The overarching objective of the VAE architecture is enabling the decoder to influence the encoder in extracting a small number of meaningful and informative latent parameters. After the training, this collaborative process ensures that the VAE identifies the bifurcation parameters from the time-series data at the VAE's information bottleneck [Sec. S2 in SI [9]).

More specifically, the encoder uses the input series x to provide a normal distribution as an output for each latent parameter z with mean μ_z and variance σ_z^2. For the training, samples of each latent parameter z are taken according to $z = \mu_z + \sigma_z\epsilon$ from the distribution $\mathcal{N}(\mu_z, \sigma_z^2)$ where $\epsilon \sim \mathcal{N}(0,1)$ is independently sampled for every training example during each training step. The decoder uses these samples along with an initial condition x_0 of the target system for predicting its time series up to an arbitrary future time T into the future. Providing the decoder with a randomly sampled initial condition ensures the focus of the encoder on deciphering the parameters characterizing the dynamics of the data rather than encoding a particular state of the system. The end-to-end training process contains: (1) a mean-squared error loss between the predicted propagation series $\hat{x}_p$ from the decoder and the target time series x, and (2) the VAE regularization term $R_z = E_1 + E_2 + E_3$, with the following meanings of the three terms. The first term $E_1 = D_{KL}[q(z,x)||q(z)p_D(x)]$ is the mutual information among the latent variables and the input data, where D_{KL} denotes the Kullback-Leibler (KL) divergence, $p_D(x)$ is the data distribution, and $q(z,x)$ is the joint distribution of the latent parameters and the data. The second term $E_2 = D_{KL}[q(z)||\prod_i q(z_i)]$ measures the total correlation among the latent parameters $[q(z) = \int dx q(z,x)$, with $q(z_i) = \int dx \Pi_{i\neq j} dz_j q(z,x)]$. The third term $E_3 = \sum_i D_{KL}[q(z_i)||p(z_i)]$ regularizes the latent space $[p(z_i) = \mathcal{N}(0,1)]$, ensuring that each latent variable follows a standard normal distribution to avoid overfitting and maintain generalizability. After the VAE is trained, the input time series x and the detected latent variable z as the control parameters are used as input for training the parameter-driven reservoir computer. For evaluation, the output prediction of the reservoir computer is monitored for any change in the dynamics and possible system collapse by changing the latent parameter beyond the VAE extracted regime [see Sec. S3 in SI [9] for details of parameter-driven reservoir computing].

The VAE regularization term R_z measures: (1) how much information about the data is captured by the latent representation (E_1), (2) the redundancy or dependencies among the different latent variables (E_2), and (3) the closeness of the distribution of each dimension of z to $\mathcal{N}(0,1)$, where the distribution parameters μ_z and σ_z^2 both are uniform across different input x (E_3). Working together, these three components force the VAE to learn a parameter representation with minimal information and independent parameters [29]. As a result, the hidden parameters can be identified through the statistics of μ_z and σ_z^2 generated by the VAE encoder for each

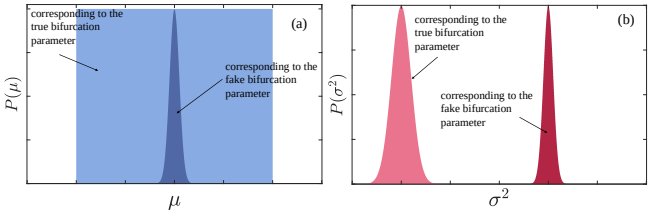

FIG. 2. Principle of VAE identification of the bifurcation parameters. The VAE designates latent-parameter channels whose number is larger than the actual number of the bifurcation parameters. With the time series data from distinct bifurcation-parameter values as the input to the VAE, each latent parameter channel produces values that follow a statistical distribution with mean μ_z and variance σ_z^2. Shown are schematic probability distributions of (a) μ_z and (b) σ_z^2, which correspond, respectively, to the true bifurcation parameter (light blue) and a "false" latent parameter that does not correspond to any actual parameter (dark blue).

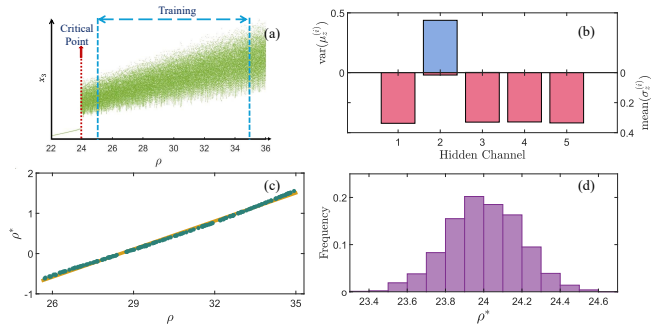

FIG. 3. Unsupervised-learning based anticipation of a critical transition in the chaotic Lorenz system with ρ as the single bifurcation parameter for $\beta = 8/3$ and $\sigma = 10$. (a) Bifurcation diagram. The VAE is trained with time series from the interval specified by the two vertical blue dashed lines. (b) VAE identification of the bifurcation parameter according to the behaviors of the variance of the mean μ_z of the latent parameter (blue) and the mean of its variance σ_z^2 (red) for the five parameter channels in the VAE. (c) VAE-detected parameter versus the ground truth physical parameter (green dots) and the linear fitting (solid yellow line). (d) Histogram of the predicted critical point ρ^* obtained from 1000 random realizations of the reservoir computer. The resulting distribution centers about the ground truth $\rho_c \approx 24.06$.

dataset. That is, a faithful latent parameter should have a high variance in μ_z and a low mean of σ_z^2, indicating that the extracted parameter is precise and informative, as illustrated in Fig. 2 with the light colors. In contrast, a parameter collapsing to the prior and failing to provide useful information will display a low variance in μ_z and high mean σ_z^2, as shown schematically in Fig. 2 with dark colors. Intuitively, a high level of variance in μ_z indicates that the latent variables have captured the key information about the data that can aid the reconstruction of the time series. However, a trivial channel with low variance in μ_z would yield almost the same μ_z for every input, regardless of variations in their underlying dynamics, and is thus uninformative. A low mean σ_z^2 ensures that the latent variables are not only informative but also precise, thereby reducing the uncertainty. This precision is regulated by both the KL divergence term E_3 and the mutual information term E_1, as uncertain latent variables would reduce the overall information captured by the VAE.

To demonstrate our unsupervised-learning scheme for anticipating critical transitions, we use simulated datasets from the chaotic Lorenz system [30] with one or two bifurcation parameters and the 1D wave system described by the Kuramoto-Sivashinsky equation [31, 32].

The Lorenz system is given by $\dot{x_1} = \sigma(x_2 - x_1)$, $\dot{x_2} = x_1(\rho - x_3) - x_2$, and $\dot{x_3} = x_1 x_2 - \beta x_3$, where σ, ρ, and β are parameters. We first consider the case where the two parameters σ and β are fixed: $\sigma = 10$ and $\beta = 8/3$, and ρ is the single bifurcation parameter. Figure 3(a) shows the bifurcation diagram of the system versus ρ. As ρ decreases, a critical transition leading to a sudden change in the dynamics of the system occurs. For $24 < \rho < 36$, the system functioning is "normal" in the sense that the system exhibits healthy oscillations. As ρ decreases, a boundary crisis [33] occurs at the critical value $\rho_c \approx 23.99$ (indicated by the red dotted line), after which the oscillations cede following a chaotic transient.

For training the VAE, we randomly sample the parameter $\rho \in [25\ 35]$, as indicated by the two blue vertical lines in Fig. 3(a). During the training, the VAE uses five latent parameter channels. Since we have only one varying bifurcation parameter, the trained VAE should only make use of one latent channel, and the rest four channels will collapse to the prior. Figure 3(b) shows the statistics of the extracted distribution of the five latent parameter channels. There is one channel with a high variance in μ_z (blue bins) and a low mean σ_z^2 (red bins), which shows that the VAE has correctly detected the number of hidden parameters. For this particular channel, the high variance in μ_z means that the VAE has captured the driving factor that alters the dynamical features in the time-series data. The low mean value of the variance σ_z^2 implies that the corresponding Gaussian distribution from sampling z is almost an impulse function with a narrow width, so we have $z \approx \mu_z$. Given these features in the μ_z and σ_z^2, the extracted μ_z can be interpreted as an estimation of the identified bifurcation parameter whose variations are responsible for the time series x from different values of the bifurcation parameter. For performance evaluation, we compare the extracted latent parameter from the VAE with the true parameter used to generate the simulated datasets, as shown in Fig. 3(c). Remarkably, the detected parameter ρ^* is linearly related to the true parameter ρ randomly chosen from the normal distribution. This linear relation suggests a one-to-one correspondence between the nontrivial latent and the ground truth parameter, as shown by the yellow fitting line in Fig. 3(c).

The VAE estimated parameter $\hat{z} = \mu_z$ and the corresponding time series data allow us to train the reservoir

computer: for each $\hat{z}$ value, training is done such that the reservoir machine can predict the state evolution of the input data for several Lyapunov times. In the testing phase, we apply a parameter change $\Delta\hat{z}$. For each resulting parameter value, we test whether the predicted attractor is the ground-truth chaotic attractor, where $\Delta\hat{z}$ can be varied systematically for the critical point ρ^* to be determined. Figure 3(d) shows a histogram of the predicted critical point ρ^*, where the linear parameter transformation $\rho^* = C_1\hat{z} + C_2$ is used with $C_1 = 4.3$ and $C_2 = 28.5$ so as to map the z values to the real domain ρ^*. Averaging over an ensemble of 1000 independent random reservoir realizations, the value of the critical point lies in the interval $\rho_c^* \approx 24 \pm 0.5$, as shown in Fig. 3(d). These results show that the parameter-driven reservoir computer, with the parameter information provided by the VAE, is capable of accurately predicting the crisis.

Note that mapping the final prediction of the transition point back to the ground-truth parameter space of ρ is solely for validating the proposed method. In an applied scenario where the ground-truth value or even the bifurcation parameter is not known, relying on predictions in the latent space of $\hat{z}$, as determined by the VAE, is sufficient for both forecasting and monitoring, as the VAE-extracted latent variable naturally takes on the role of an effective bifurcation parameter.

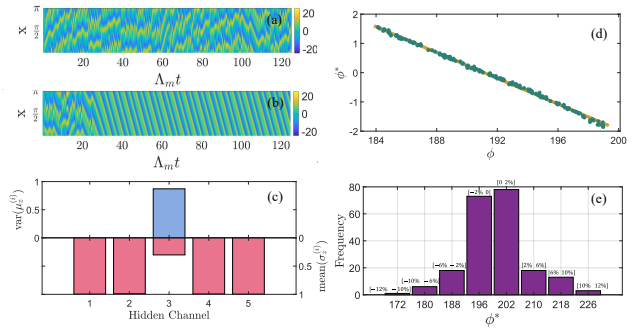

FIG. 4. Predicting critical transition in the nonlinear wave system described by the Kuramoto-Sivashinsky equation. Two examples of sustained and transient spatiotemporal chaos are shown for (a) $\phi = 199.94 \leq \phi_c$ and (b) $\phi = 200.14 \geq \phi_c$. (c) Variance of μ_z (blue) and mean of σ_z^2 (red) for the five parameter channels in the trained VAE. (d) VAE's detected parameters versus the ground truth parameter (green dots) and linear fitting (yellow solid line). (e) Histogram of the predicted critical point ϕ^* and the relative errors with respect to the ground truth $\phi_c \approx 200.04$. The distribution is collected from 210 random reservoir realizations.

We next study the spatiotemporal system described by the Kuramoto-Sivashinsky equation [31, 32]: $\partial_t u = -\nu\partial_x^4 u - \phi(\partial_x^2 u + u\partial_x u)$, where $u(t, x)$ is the dynamical variable at position x and time t, $\nu = 4$ is the damping coefficient and ϕ is a bifurcation parameter. The spatial domain is $0 \leq x \leq \pi$ with the periodic boundary condition. A critical transition occurs at $\phi_c \approx 200.04$, where there is sustained and transient spatiotemporal chaos for $\phi < \phi_c$ and $\phi > \phi_c$, respectively. Two examples are shown in Figs. 4(a) and 4(b), respectively. For training the VAE, we randomly sample 500 points in $\phi \in [184\ 199]$ that belongs to the regime of sustained spatiotemporal chaos. During the training, the VAE uses five latent parameter channels. Figure 4(c) shows that only one channel exhibits a high variance in μ_z and a low mean σ_z^2, indicating that the unsupervised learning scheme has correctly identified the hidden parameter. Figure 4(d) shows the extracted latent parameters $\hat{z}$ in comparison with the ground truth parameters ϕ for generating the datasets (represented by the green points). The results suggest the existence of a linear transformation mapping the estimated parameter $\hat{z}$ to the real values ϕ, as indicated by the yellow line in Fig. 4(d). The detected parameter z along with their corresponding time-series data are then used to train the parameter-driven reservoir computer. To make predictions, a new control parameter value is generated with a small change $\Delta\hat{z}$, and the output of the reservoir computer is evaluated to determine whether the predicted state corresponds to sustained or transient chaos. Figure 4(e) shows a histogram of the predicted critical point ϕ^*, where the linear transformation $\phi^* = C_1\hat{z} + C_2$ with $C_1 = -4.5$ and $C_2 = 191.13$ is used to bring back the z to the real parameter ϕ^*. The histogram reveal that about 57% of the predicted values of ϕ^* have relative error $|\delta| < 2\%$. For the accuracy relaxed to $|\delta| < 10\%$, the fraction of correct prediction becomes 98%. These results indicate that our unsupervised learning framework is applicable to anticipating critical transitions in spatiotemporal chaotic systems.

Additional tests have been performed for the Lorenz system with two bifurcation parameters [Sec. S4 in SI [9]]. The issue of partial state observation is addressed [Sec. S5 in SI [9]] and a chaotic ecosystem violating the sparsity condition has also been tested [Sec. S6 in SI [9]]. For all the systems tested, the selection of the optimal hyperparameters is described in Sec. S7 in SI [9].

To summarize, our unsupervised learning framework integrates VAE with parameter-driven reservoir computing, where VAE is used to identify the key bifurcation parameter(s) and its (their) variations. The extracted parametric information is fed into a parameter-driven reservoir computer for anticipating critical transitions. The framework was tested using diverse examples in different settings. In all cases, the robustness and accuracy in anticipating critical transitions were demonstrated. Our work advocates the use of modern machine learning methods in predicting the future behaviors of nonlinear and complex dynamical systems in real-world scenarios, with potential applications in different fields.

This work was supported by AFOSR under Grant No. FA9550-21-1-0438 and by ARO under Grant No. W911NF-24-2-0228. L.-W.K. was supported by the Eric and Wendy Schmidt AI in Science Postdoctoral Fellowship, a Schmidt Futures Program.

Appendix A: Previous approaches to anticipating critical transitions and limitations

Previously, anticipating critical transitions leading to a system collapse was done by using the sparse optimization approach to finding the governing equations of the target dynamical system [10] or by exploiting reservoir-computing based machine learning [13]. Both approaches have limitations: the former requires that the system equations have a sparse structure and the latter necessitates information about the bifurcation parameter whose variations lead to a critical transition. The machine-learning approach can thus be categorized as supervised learning. The main motivation for our work is the real-world situations where nothing about the system parameters is known and the only available information is some time-series data observed or measured from a number of dynamical variables of the system. To anticipate a critical transition in such a case, the parameter variations need to be "learned" from the data without any "teacher," rendering unsupervised the learning process. The main contribution of our work is an unsupervised learning framework tailored to anticipating critical transitions induced by parameter changes but no prior information about any parameter of the system is needed.

1. Sparse optimization for finding the governing equations from data

An earlier approach to anticipating critical transitions was articulated [10] in 2011, which is based on finding the governing equations of the system from data [34–36]. The idea is that the equations of certain nonlinear dynamical systems possess a sparse structure: they contain a small number of terms expressible in terms of elementary mathematical functions, such as power-series or Fourier-series terms. Well studied, paradigmatic systems in nonlinear dynamics such as the Lorenz system [30], the Rösller system [37], the Hénon map [38], and the standard map belong to this category. For such systems with a "simple" equation structure, the problem of finding the governing equations becomes that of finding the coefficients associated with these terms in a power series or a Fourier series, which can be solved by the standard sparse-optimization methods such as compressive sensing [39–43]. This equation-finding approach was extended to solving a variety of problems in nonlinear systems and complex networks [44, 45] such as unveiling the structure of complex oscillator networks [46] and social networks [47], data-driven forecasting of synchronizability of complex networks [48], detection of hidden nodes [49, 50], and reconstruction of propagation or spreading networks [51].

The sparsity condition required for this approach is in fact self-sabotage: while it is the reason that the powerful sparse-optimization methods are applicable, it also represents a condition that many real-world systems do not meet. As such, this approach is limited to systems whose equation structures are sparse. There were also previous works on anticipating a common class of critical transitions known as tipping at which a "normal" stable steady state of the system is destroyed and replaced by another one that can be catastrophic (e.g., population extinction in an ecosystem) [7, 52–60] through detecting early-warning signals [12, 61–68]. These methods are based on the observation that, as the system approaches a tipping point, the statistical fluctuations of the dynamical variables would increase dramatically, as the leading eigenvalue associated with the stable steady state is about to approach zero.

2. Limitation of previous parameter-driven reservoir computing

A limitation of parameter-driven reservoir computing is that it requires the knowledge of the bifurcation or control parameter that may change with time. In particular, during the standard training process [13], the bifurcation parameter values need to be injected into the neural network together with the state time series of the target system. If the exact parameter values are not known, the relative trend of the parameter may also suffice. For instance, suppose we have four different training data sequences, such as the population time series of some animal species collected from four different locations labeled as A, B, C, and D, respectively. Suppose there is one dominant parameter, the average temperature, which makes the population dynamics different among the four places. Then, to make any interpolation or extrapolation possible, we at least need to know the order (descending or ascending) of the parameter among the four data sets, be it A-B-C-D or A-C-B-D, for example. With different orders, we could end up with qualitatively different results, especially if we perform an extrapolation, as we may have the parameter tendency qualitatively wrong. Overcoming this difficulty requires identifying uncontrolled dynamical parameters that cause the variations in the dynamics of a set of observed trajectories. This ensures interpretability [69] without compromising the flexibility inherent in machine learning. Simply learning a black-box predictor for each trajectory does not provide an accurate extrapolation of the dynamics' changes among different trajectories [70].

It was demonstrated that the parameters governing the dynamics of a complex nonlinear system can be encoded in the learned readout layer of a reservoir computer [71]. Another approach is a variational-autoencoder (VAE) architecture consisting of an encoder that extracts the physical parameters characterizing the system dynamics and a decoder that acts as a predictive model and propagates an initial condition forward in time given the extracted parameters [29]. It is worth noting that VAEs are widely used for dimensionality reduction and unsupervised learning tasks [72] and for studying a wide variety

of physical phenomena.

Appendix B: Variational autoencoders (VAEs)

VAEs [73, 74] are a generative model in unsupervised learning with the advantages of learning intricate and continuous latent representations of complex datasets. VAEs combine the principles from variational inference with deep neural networks to efficiently capture the underlying structure of the input data. The core of a VAE architecture is an encoder-decoder framework, as shown in Fig. 5(a). The encoder part can be regarded as a stochastic process that samples a representation $z \sim Q_\phi(z|x)$ of the input data x from a distribution $Q_\phi(z|x)$ parameterized by the functions of the input, where ϕ represents the encoder parameters. This distribution typically follows a Gaussian form with mean μ and standard deviation σ. More specifically, the encoder first yields the distribution parameters $(\mu, \sigma) = q_\phi(x)$ for each input data x. Then the sampling process is denoted as $z = \mu + \sigma\epsilon$, where ϵ is drawn from a standard normal distribution. The decoder part, with a generative process that can be formulated as $\hat{x} \sim P_\theta(\hat{x}|z)$, is responsible for the task of reconstructing the input data x from the sampled latent variable z, with θ representing the decoder parameters.

The training goal of a VAE is maximizing the evidence lower bound (ELBO), which is a variational approximation of the logarithmic-likelihood of the data. The ELBO encompasses two key components: a reconstruction term and Kullback-Leibler (KL) divergence between the approximate posterior and the prior distributions of the latent variables, where the former ensures accurate data reconstruction and the latter regularizes the learned latent space, preventing it from deviating excessively from a predefined prior distribution (often a unit Gaussian). The training of a VAE is thus a minimax optimization problem - maximizing the reconstruction fidelity while minimizing the channel capacity in the bottleneck. The overall loss function can be written as

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 + \mathrm{KL}\Big(q_\phi(z|x) \parallel p(z)\Big), \quad \text{(B1)}$$

where x_i and $\hat{x}_i$ are the i^{th} data point and the reconstructed point, respectively. Here, the second term is the regularization term R_z. It is equivalent to the expression in the main text where R_z is decomposed into three terms $R_z = E_1 + E_2 + E_3$. While the decomposed expression enables an intuitive understanding, the expression in Eq. (B1) facilitates numerical calculation. The optimization of the VAE parameters with respect to this loss function can be achieved through backpropagation. Overall, the advantages of VAEs are modeling complex data distributions and generating novel samples from the learned latent space, making them particularly well-suited for diverse applications such as data generation and dimensionality reduction. A special feature of VAEs that is particularly relevant to our work is their ability to extract informative latent parameters from time-series data. *When combined with parameter-driven reservoir computing, the framework makes it possible to anticipate critical transitions even without direct knowledge about the bifurcation parameter, as its variations can be extracted solely from the time series data.*

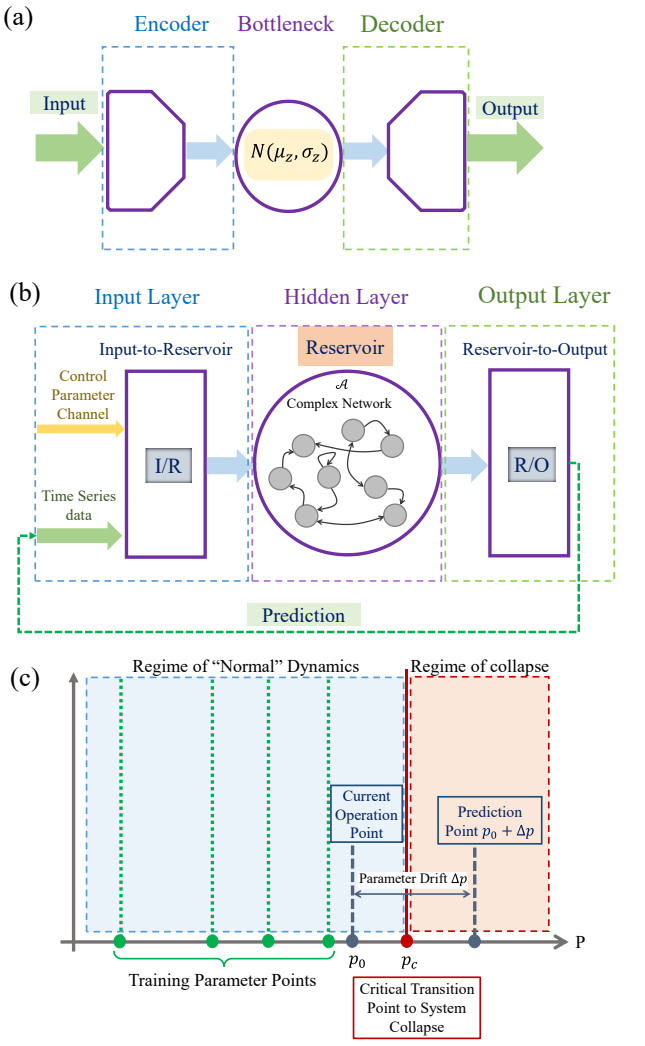

FIG. 5. VAEs and parameter-driven reservoir computing. (a) Structure of a VAE. (b) Structure of parameter-driven reservoir computing, where the input consists of two components: time-series data and the values of the VAE-extracted bifurcation parameter. (c) Scheme of training for predicting critical transitions. The light blue region represents the regime of normal system operation while the light orange region denotes the parameter regime of collapse. A critical transition from normal operation to collapse occurs at the parameter value p_c, and $p_0 < p_c$ is the current operation point. Historical time-series data from a small number of parameter values in the normal regime are used for training, as indicated by the four vertical dashed green lines. Prediction is done for $p = p_0 + \Delta p$, where $\Delta p > 0$ is a parameter drift.

Appendix C: Parameter-driven reservoir computing

Reservoir computing is a class of recurrent neural networks [18, 19, 75] capable of self dynamical evolution. The core of reservoir computing is a hidden layer that contains a complex network of mutually interacting nonlinear neurons. The architecture consists of an input layer, the reservoir network in the hidden layer (the recurrent layer), and an output layer. The reservoir network and the input layer are both randomly predetermined and fixed throughout the training and testing phases; Only the output layer is optimized according to the training data and the loss function, where a ridge regression is usually sufficient for the optimization. The training process is thus finding a linear combination of the nonlinear response signals generated in the reservoir network.

Parameter-driven reservoir computing has an input parameter channel in addition to the data input channels, as shown in Fig. 5(b), which enables each neuron in the network to receive specific parameter values associated with input time-series data [13], making the reservoir computer "aware" of the parameter changes in the target system. The training process is illustrated in Fig. 5(c), where p is the bifurcation parameter of the target system. As p varies, a critical transition occurs at p_c that separates the regime normal system functioning ($p < p_c$, light blue) and collapse after a transient ($p > p_c$, light orange). Assume that p increases slowly with time and let p_0 be the parameter value of the current system operation. The training is done using the historical time series from a small number of distinct parameter values, all to the left of p_0 in the normal regime, as indicated by the four vertical dashed green lines in Fig. 5(c). For each of these parameter values, training is done such that the reservoir computer generates accurate short-term prediction of the system's state evolution. This adaptive training paradigm empowers the reservoir computer with the ability to anticipate critical transitions induced by parameter variations. In particular, to predict a potential system collapse in the future, we assume a parameter change $\Delta p > 0$ and let the reservoir computer generate the dynamical behavior at the future parameter value $p_0 + \Delta p$ through self evolution. For $p_0 + \Delta p < p_c$, a well trained reservoir computer should generate the normal attractor of the target system. However, for $p_0 + \Delta p > p_c$, the reservoir computer should generate an attractor that is indicative of system collapse.

Appendix D: Unsupervised learning for anticipating critical transition of Lorenz system with two independent bifurcation parameters

For a real-world system, the presence of multiple bifurcation parameters can be expected. Can our unsupervised learning framework be effective for anticipating a critical transition in such a case? As a concrete example, we consider the Lorenz system with ρ and β as the two independent bifurcation parameters.

Figure 6(a) shows a 2D bifurcation diagram of the system with respect to the parameters ρ and β. For parameters in the blue region, the system functioning is "normal" in the sense that its dynamical behavior exhibits "healthy" oscillations. The points belonging to the yellow region correspond to an abnormal behavior of the system. These two regions are separated by a critical curve (a one-dimensional set of critical points) as indicated by the red dotted line. Near these points, the variation of parameters leads to a sudden change in the dynamics triggered by a boundary crisis [33].

For training the VAE we randomly sample 500 points with $\rho \in [25\ 35]$ and $\beta \in [0.8\ 2.66]$, as indicated by the blue rectangle in Fig. 6(a). Similar to the case of a single bifurcation parameter, during the training, the VAE uses five latent parameter channels. The number of hidden parameters is determined using the statistics of the extracted distribution parameters μ_z and σ_z^2 for each dataset. In particular, Fig. 6(b) shows the statistics of the extracted distribution of the five latent parameter channels. There are two channels with a high variance in μ_z and a low mean σ_z^2, indicating that the unsupervised learning framework has correctly detected the number of hidden parameters. To evaluate the method's performance, we plot the extracted latent parameters (z_1 and z_2) from the learning scheme (shown as points) against the true physical parameters (ρ and β) used for generating the simulated datasets, as depicted in Fig. 6(c), where dark (light) purple points correspond to the first (second) latent parameter. The results indicate that the latent parameters can be interpreted as a linear combination of the ground truth parameters, where the shaded surfaces represent the results of the linear combination of the actual ground truth parameters.

Using the VAE's detected parameter and their corresponding time series data, we train the parameter-driven reservoir computer. For each pair of the detected parameters, training is done such that the reservoir machine can predict the state evolution of the input data for several Lyapunov times. After the training, we apply parameter changes Δz_1 and Δz_2 in a systematic fashion by dividing the parameter plane into two sets of potential testing points. The first set comprises points with healthy behavior but lying outside the training rectangle, as depicted by the light blue points in Fig 6(a). The second set consists of points from the yellow region in Fig 6(a), which generate an abnormal behavior in the sense of lack of oscillations in the long-term dynamics. For each resulting parameter value, we assess whether the predicted system state represents a chaotic attractor. The outcomes of one iteration of the reservoir computer are shown in Fig. 6(d), where the green and red dots represent the predictions of the healthy and unhealthy behavior, respectively. The red (green) points within the first (second) set indicate an unhealthy (healthy) behavior wrongly labeled as healthy (unhealthy). For illustrative purposes, we apply a linear

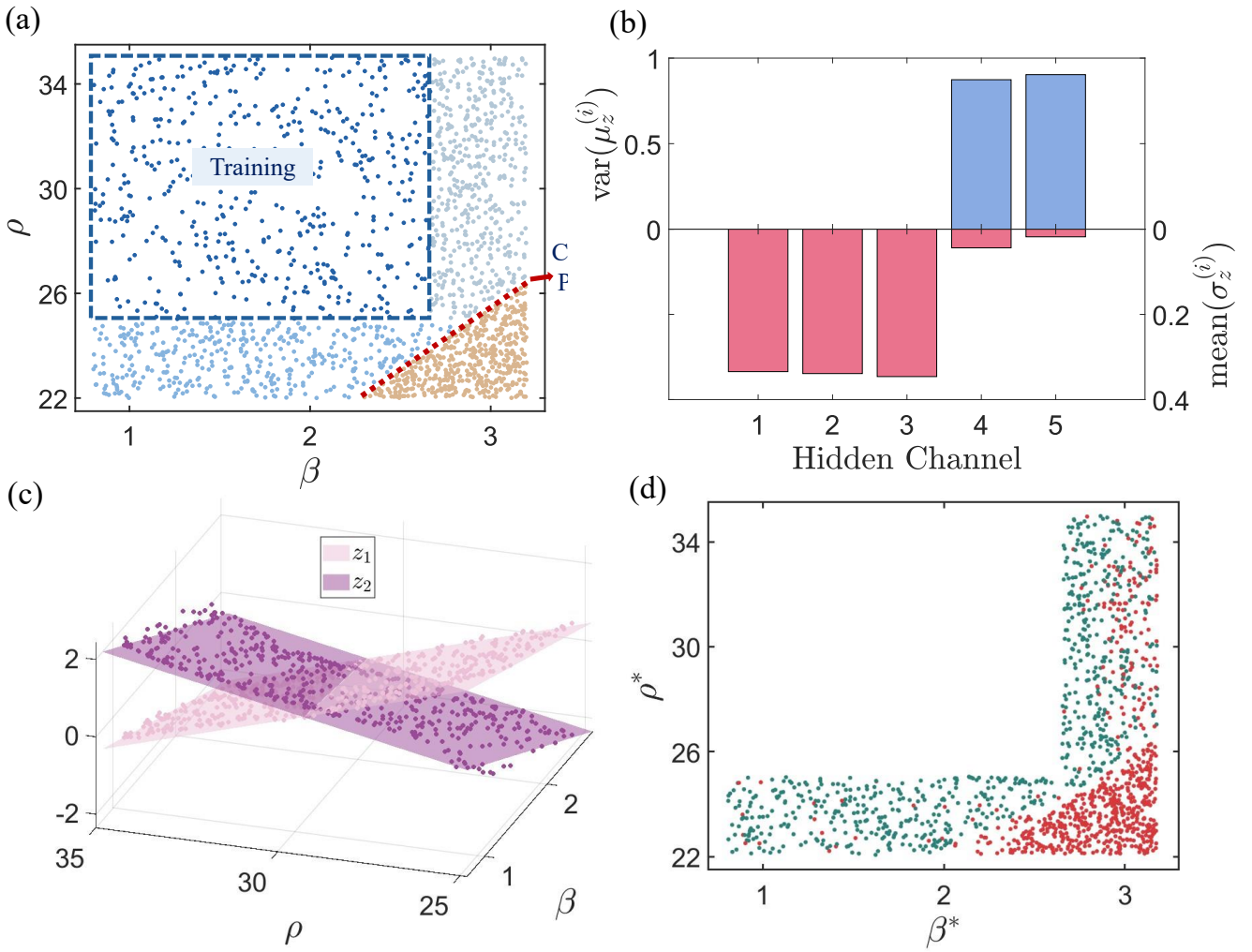

FIG. 6. Unsupervised learning for anticipating critical transition of Lorenz system with two independent bifurcation parameters ρ and β. (a) 2D bifurcation diagram with respect to the parameter plane (ρ, β) for $\sigma = 10$. The VAE is trained with time series taken within the blue regime in which the system operates normally. The training parameter values is indicated by the light blue dashed lines. (b) Identification of the bifurcation parameters from the variance of the mean μ_z (blue) and the mean of the variance σ_z^2 (red) for the five parameter channels in the VAE. (c) VAE's detected parameter values in comparison with the ground truth physical parameters, where the dark and light purple dots corresponding to the first and second extracted latent parameter, respectively. Each shaded surface represents a linear combination of the ground truth parameters (dark and light purple). (d) Predicted parameter points transformed into the real domain. The red dots represent points with unhealthy behaviors where the reservoir computer output exhibits a critical transition, while the green red dots denote healthy behaviors - those without any critical transition.

transformation:

$$[\rho^* \ \beta^*]^T = C[z_1, z_2, 1]^T,$$

where

$$C = \begin{pmatrix} 1.74 & -2.45 & 30.15 \\ -0.48 & -0.33 & 1.72 \end{pmatrix},$$

to bring the detected parameters z_1 and z_2 to their real values ρ^* and β^*. To assess the prediction performance, we compute the false positive rate (FPR) and a false negative rate (FNR), where the former represents the fraction of the red points in the first set and the latter is the fraction of the green points in the second set. We obtain FPR $\approx 23\%$ and FNR $\approx 1\%$. A low FNR is desirable because it indicates a small miss rate, suggesting that our unsupervised learning framework is capable of predicting the critical bifurcation with small errors even with two independent bifurcation parameters.

Appendix E: Anticipating critical transition with partial state observation

The results so far presented are obtained under the assumption that the bifurcation parameter of the target system is inaccessible but the time series from all the dynamical variables of the target system can be obtained, corresponding to full state observation. In real applications, it may occur that only a subset of the dynamical variables can be observed or measured, a situation referred to as partial state observation. Is our unsupervised learning framework still able to predict critical transitions?

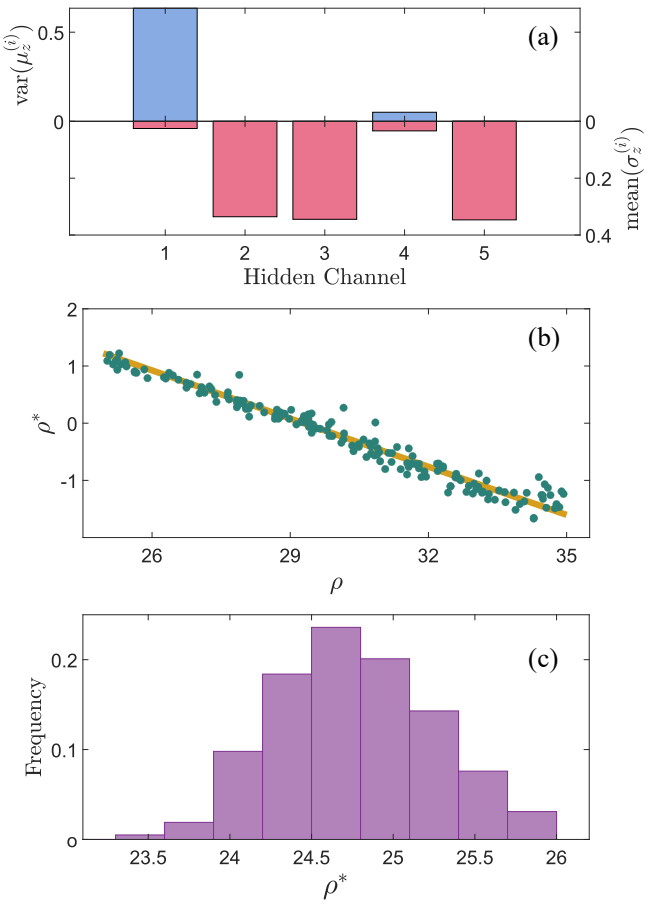

FIG. 7. Unsupervised-learning based prediction critical transition in the Lorenz system with partial state observation. (a) Variance of μ_z (blue) and mean of σ_z^2 (red) for the five parameter channels in the VAE. (b) VAE's detected parameters versus the ground truth physical parameters (green dots) and the linear fitted line (solid yellow line). (c) Histogram of the predicted critical point ρ^*, with 1000 random realizations of the reservoir computer.

In nonlinear dynamics, the well-established methodology to deal with partial state observation is the classical Takens' delayed-coordinate embedding framework [76], where the phase space of the system of interest can be reconstructed even with a single time series. We follow this principle by changing the structure of the decoder. Recall that, in the case of full state observation, the decoder takes only one time step of state $\hat{x}_i$ to predict the next $\hat{x}_{i+1}$. With partial state observation, we apply time-delay embedding in the input of the decoder to perform the reconstruction/prediction task. Specifically, the input to the decoder now contains several steps of the (partial) state variables $\{\hat{x}_{i-d}, \hat{x}_{i-d+1}, ..., \hat{x}_{i-1}, \hat{x}_i\}$. The output of the decoder is still the next-step prediction $\hat{x}_{i+1}$.

To give a concrete example, we again use the chaotic Lorenz system, assuming that only the first variable (x) can be measured and the single bifurcation parameter is ρ. Figure 7(a) shows the statistical characteristics of the extracted distribution across five latent parameter channels. Notably, there is only one channel with a substantial variance in μ_z alongside a minimal mean σ_z^2, indicating that an accurate identification of the number of hidden parameters has been achieved. Figure 7(b) shows the extracted parameter values in comparison the true physical parameter values (green dots). A comparison between the result in Fig. 7(b) with that in Fig. 3(c) for the full-state observation case indicates that, while there is information loss due to having access to only partial state observation, it is still possible to find a linear transformation (the solid yellow line) to map the detected latent parameter z to the true parameter ρ.

We can now train the reservoir computer using the detected parameter $\hat{z}$ and their corresponding time series data. Systematically applying small parameter change $\Delta\hat{z}$ to check if the long-term dynamics are generated by a chaotic attractor, we obtain the critical point ρ^*, as shown by the histogram in Fig. 7(c), where the linear transformation $\rho^* = C_1\hat{z} + C_2$ with $C_1 = -3.5$ and $C_2 = 29.3$ is used to bring back the $\hat{z}$ to real-domain ρ^*. Comparing the histogram with that in Fig. 3(d) for the case of full state observation reveals that, despite the adverse effects of information loss associated with partial state observation, our unsupervised learning framework remains capable of predicting the critical transition.

Appendix F: Anticipating critical transition in a chaotic ecosystem that violates the sparsity condition

The examples treated in the main text belong to the category of dynamical systems with a sparse structure. Here we consider an example in ecosystems in which the sparsity condition is violated in the sense that the power-series expansions of the component functions of the velocity field contain an infinite number of terms, and demonstrate that a typical critical transition can be faithfully anticipated by our unsupervised learning framework.

There is significant concern about human-activity induced environmental changes and how these changes may lead to catastrophic event such as sudden species extinction. In a typical ecological system, there are two coexist-

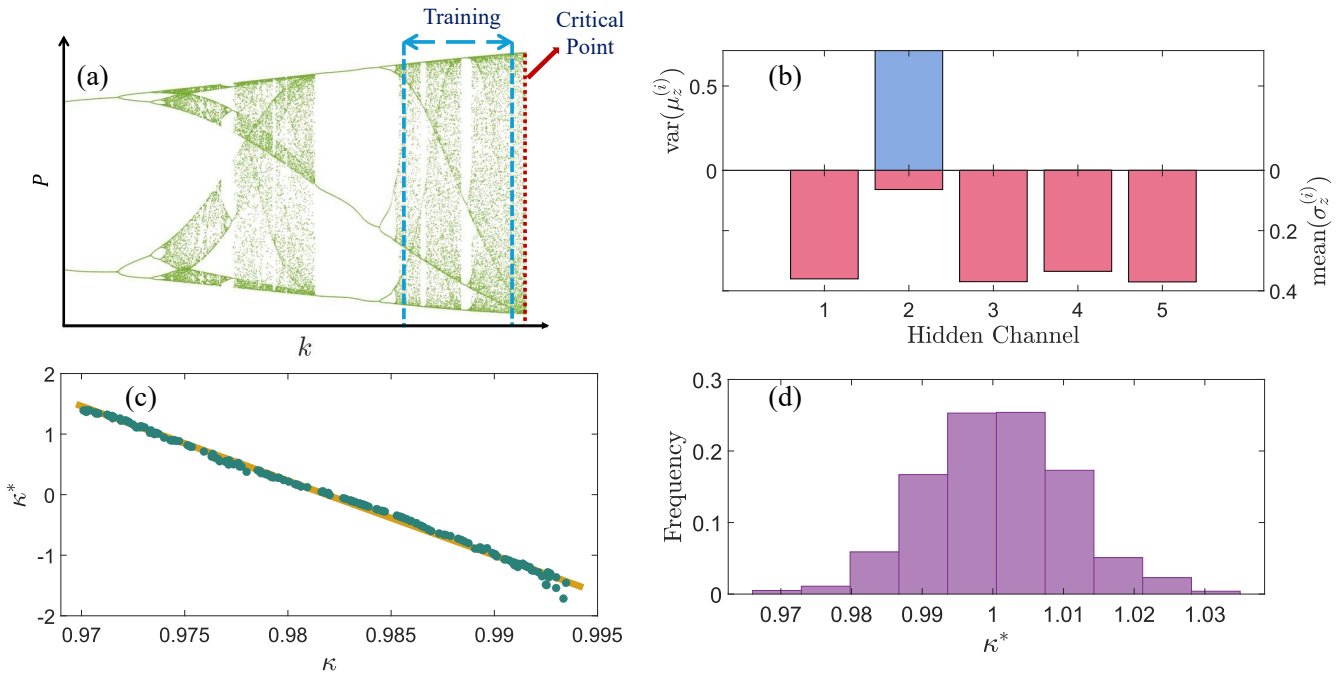

FIG. 8. Anticipating a critical transition in the chaotic food-chain system (F1). (a) Bifurcation diagram with κ. The fixed values of the other parameters are: $x_c = 0.4$, $y_c = 2.009$, $x_p = 0.08$, $y_p = 2.876$, $R_0 = 0.16129$, and $C_0 = 0.5$. The VAE is trained with time series taken from the blue interval in the green regime in which there is healthy coexistence of all three species. The training interval is indicated by the two vertical blue dashed lines. (b) Variance of μ_z (blue) and mean of σ_z^2 (red) for the five parameter channels in the VAE. (c) VAE's detected parameters versus the ground truth physical parameters (green dots) and the linear fitted line (solid yellow line). (d) Histogram of the predicted critical point from 1000 random realizations of the reservoir machine. The distribution centers around the ground-truth critical point $K_c = 1.00$.

ing states: one associated with healthy survival and the other with extinction. Changes in the environment can change the parameters of the system, triggering a critical shift that occurs when the system reaches a critical point, resulting in the disappearance of the survival state and leaving only extinction as the final state. To demonstrate that our unsupervised learning framework can anticipate critical transitions in ecosystem, we consider the chaotic food chain system of three species [77]:

$$\begin{aligned}\dot{R} &= R\Big(1-\frac{R}{\kappa}\Big)-\frac{x_c y_c C R}{R+R_0},\\ \dot{C} &= x_c C\Big(\frac{y_c y_c R}{R+R_0}-1\Big)--\frac{x_p y_p P C}{C+C_0},\\ \dot{P} &= x_p P\Big(\frac{y_p C}{C+C_0}-1\Big),\end{aligned} \tag{F1}$$

where the dynamical variables R, C, and P are the resource, consumer, and predator species density, respectively, κ is a parameter characterizing the environmental capacity of the resource species and is a bifurcation parameter reflecting the effects of environment changes on the system. Other parameters of the system $(x_c, y_c, x_p, y_p, R_0$, and $C_0)$ are fixed. Figure 8(a) shows the bifurcation diagram of the system (F1).

The procedures for generating training data closely resemble those for the chaotic Lorenz system. Here, we randomly sample the parameter $\kappa \in [0.97\ 0.994]$, as indicated by the two blue vertical lines in Fig. 8(a). For each parameter value, we integrate Eq. (F1) to generate the training time series. Figure 8(b) shows the statistical characteristics of the extracted distribution across five latent parameter channels. It can be seen that there is only one channel with a large variance in μ_z and low mean σ_z^2. Figure 8(c) shows the VAE extracted latent parameters in comparison with the true physical parameter (green dots). There is a linear transformation to map the detected parameter z to the true parameter κ (yellow solid line). The parameter-driven reservoir computer is trained using the detected parameter z and their corresponding time series data. To find a possible critical transition, a small parameter change Δz is applied to the trained reservoir computer to check if it produces a survival chaotic attractor. Figure 8(d) shows a histogram of the predicted critical point κ^*, where the linear transformation $\kappa^* = C_1\hat{z}+C_2$ is used to bring back the extracted z to the real κ^*. The result in Fig. 8(d) indicates that our unsupervised learning framework can accurately predict the critical transition in the chaotic food-chain system.

TABLE I. Hyperparameter configurations for each example in the main text: Sets A-E correspond to, respectively, chaotic Lorenz system with a single control parameter, chaotic Lorenz system with two control parameters, chaotic Lorenz system with partial state observation, spatiotemporal Kuramoto-Sivashinsky system, and the chaotic food chain system.

Hyperparameters		Set A	Set B	Set C	Set D	Set E
VAE	n_b	100	100	100	50	100
	lr	2×10^{-3}	2×10^{-3}	2×10^{-3}	1×10^{-3}	5×10^{-3}
	β_{VAE}	1×10^{-3}	1×10^{-3}	1×10^{-3}	1	1×10^{-3}
Parameter Driven Reservoir Computing	n_r	800	800	800	4000	900
	d	550	730	580	1902	4
	λ	1.34	0.55	0.63	0.85	2.3
	k_{in}	0.04	0.08	0.09	0.01	3.6
	k_b	3	0.23	1.6	2.78$	0.5
	b_0	3.5	-1.9	4.4	1.31	-2.2
	α	0.7	0.15	0.17	0.91	0.3

Appendix G: Hyperparameter Optimization

In machine learning, the hyperparameters are those defining the neural-network architecture and those associated with the training process, which play an important role in achieving accurate testing and generalization results. Determining the optimal values of the hyperparameters is computationally demanding and is often implemented as a black-box optimization problem. Earlier methods included grid search based on an exhaustive evaluation of all possible hyperparameter combinations within a constrained search space [78], which are infeasible for high-dimensional problems due to the computational cost and lack of scalability. Random search offers a more efficient alternative but still suffers from high computational expenses and lacks adaptability [79]. Reinforcement learning has been exploited for finding the appropriate architecture of CNNs and for determining training parameters [80], but a deficiency is the absence of a global exploration strategy. Bayesian methods, while sophisticated, provide an appealing alternative by constructing a probabilistic model over an objective function, utilizing techniques such as Gaussian processes or random forests [81, 82]. We employ the Bayesian optimization method (implemented using the *skopt*package in Python), for both the VAE and parameter-driven reservoir computing.

A VAE, similar to any deep neural network, typically contains various hyperparameters including the number of hidden layers, choice of activation function, optimizer selection, batch size (n_b), learning rate (lr), and the regularization parameter (β_{VAE}). In our work, the number of hidden layers is fixed at 32, the activation function is chosen to be the ReLU, and the Adam optimization algorithm is used. The remaining hyperparameters that need to be optimized are the batch size, learning rate, and the regularization parameter. It is worth noting that for the task of parameter identification, the regularization parameter is particularly important, as it can affect the extraction of parameters. An optimal choice of the regularization parameter can facilitate the learning of disentangled representations. Finding an optimized β_{VAE} value, however, demands a delicate balance, as excessively high values yield irrelevant parameters while overly low values will compromise independence enforcement.

For parameter-driven reservoir computing, the hyperparameters [13] consist of the predefined parameters characterizing the input architecture and the neural network in the hidden layer, which include the network size (n_r), the average degree (d), the spectral radius (λ), the scaling factor of the elements of the data input matrix (k_{in}), the parameters associated with the parameter-input matrix, and the leakage parameter (α). The optimization strategy entails training multiple reservoir computers for each hyperparameter set, computing the average validation root-mean-square error, and integrating this feedback into the Bayesian algorithm. Iterative training with various reservoir realizations mitigates the error fluctuations. With a few hundred Bayesian iterations, hyperparameter values yielding the lowest validation root-mean-square error across all iterations are selected, prioritizing the overall performance over specific iteration outcomes. The results of hyperparameter optimization for the examples in our work are listed in Table I.

[1] A. Hastings, K. C. Abbott, K. Cuddington, T. Francis, G. Gellner, Y.-C. Lai, A. Morozov, S. Petrovskii, K. Scranton, and M. L. Zeeman, Transient phenomena in ecology, Science **361**, eaat6412 (2018).

[2] T. M. Lenton, J. Rockström, O. Gaffney, S. Rahmstorf, K. Richardson, W. Steffen, and H. J. Schellnhuber, Climate tipping points—too risky to bet against, Nature **575**, 592 (2019).

[3] I. Dobson and H.-D. Chiang, Towards a theory of voltage collapse in electric power systems, Sys. Cont. Lett. **13**,

253 (1989).
[4] M. Scheffer, *Critical Transitions in Nature and Society*, Princeton Studies in Complexity (Princeton University Press, 2020).
[5] S. M. O'Regan, E. B. O'Dea, P. Rohani, and J. M. Drake, Transient indicators of tipping points in infectious diseases, J. R. Soc. Interface. **17**, 20200094 (2020).
[6] E. H. van Nes, B. M. Arani, A. Staal, B. van der Bolt, B. M. Flores, S. Bathiany, and M. Scheffer, What do you mean, 'tipping point'?, Trends Ecol Evol. **31**, 902–904 (2016).
[7] M. Scheffer, J. Bascompte, W. A. Brock, V. Brovkin, S. R. Carpenter, V. Dakos, H. Held, E. H. Van Nes, M. Rietkerk, and G. Sugihara, Early-warning signals for critical transitions, Nature **461**, 53 (2009).
[8] M. Scheffer, S. R. Carpenter, T. M. Lenton, J. Bascompte, W. Brock, V. Dakos, J. van de Koppel, I. A. van de Leemput, S. A. Levin, E. H. van Nes, M. Pascual, and J. Vandermeer, Anticipating critical transitions, Science **338**, 344–348 (2012).
[9] Supplementary Information provides details elaborating the results in the main text. It is helpful but not essential for understanding the main results of the paper. It contains the following: (1) background materials on previous approaches to anticipating critical transitions and limitations, (2) a detailed description of variational autoencoders, (3) a description of parameter-driven reservoir computing, (4) results on unsupervised learning for anticipating critical transition of Lorenz system with two independent bifurcation parameters, (5) results on anticipating critical transition with partial state observation, (6) results on anticipating critical transition in a chaotic ecosystem that violates the sparsity condition, and (7) hyperparameter Optimization.
[10] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi, Predicting catastrophes in nonlinear dynamical systems by compressive sensing, Phys. Rev. Lett. **106**, 154101 (2011).
[11] S. H. Lim, L. Theo Giorgini, W. Moon, and J. S. Wettlaufer, Predicting critical transitions in multiscale dynamical systems using reservoir computing, Chaos **30** (2020).
[12] T. M. Bury, R. Sujith, I. Pavithran, M. Scheffer, T. M. Lenton, M. Anand, and C. T. Bauch, Deep learning for early warning signals of tipping points, Proc. Natl. Acad. Sci. (USA) **118**, e2106140118 (2021).
[13] 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).
[14] 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).
[15] H. Fan, L.-W. Kong, Y.-C. Lai, and X. Wang, Anticipating synchronization with machine learning, Phys. Rev. Res. **3**, 023237 (2021).
[16] L.-W. Kong, H. Fan, C. Grebogi, and Y.-C. Lai, Emergence of transient chaos and intermittency in machine learning, J. Phys. Complex. **2**, 035014 (2021).
[17] 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).
[18] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks-with an erratum note, GMD **148**, 13 (2001).
[19] 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).
[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] 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).
[22] D. J. Gauthier, E. Bollt, A. Griffith, and W. A. Barbosa, Next generation reservoir computing, Nat. Commun. **12**, 5564 (2021).
[23] Z.-M. Zhai, L.-W. Kong, and Y.-C. Lai, Emergence of a resonance in machine learning, Phys. Rev. Res. **5**, 033127 (2023).
[24] L.-W. Kong, G. A. Brewer, and Y.-C. Lai, Reservoir-computing based associative memory and itinerancy for complex dynamical attractors, Nat. Commun. **15**, 2815 (2024).
[25] M. Yan, C. Huang, P. Bienstman, P. Tino, W. Lin, and J. Sun, Emerging opportunities and challenges for the future of reservoir computing, Nat. Commun. **15**, 2056 (2024).
[26] R. Xiao, L.-W. Kong, Z.-K. Sun, and Y.-C. Lai, Predicting amplitude death with machine learning, Phys. Rev. E **104**, 014205 (2021).
[27] 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).
[28] S. Panahi, L.-W. Kong, M. Moradi, Z.-M. Zhai, B. Glaz, M. Haile, and Y.-C. Lai, Machine-learning prediction of tipping, arXiv preprint arXiv:2402.14877 (2024).
[29] P. Y. Lu, S. Kim, and M. Soljačić, Extracting interpretable physical parameters from spatiotemporal systems using unsupervised learning, Phys. Rev. X **10**, 031056 (2020).
[30] E. N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. **20**, 130 (1963).
[31] Y. Kuramoto, Diffusion-induced chaos in reaction systems, Prog. Theo. Phys. Supp. **64**, 346 (1978).
[32] G. I. Sivashinsky, On flame propagation under conditions of stoichiometry, SIAM J. Appl. Math. **39**, 67 (1980).
[33] C. Grebogi, E. Ott, and J. A. Yorke, Crises, sudden changes in chaotic attractors, and transient chaos, Physica D **7**, 181–200 (1983).
[34] J. P. Crutchfield and B. McNamara, Equations of motion from a data series, Complex Sys. **1**, 417 (1987).
[35] E. M. Bollt, Controlling chaos and the inverse frobenius-perron problem: global stabilization of arbitrary invariant measures, Int. J. Bif. Chaos **10**, 1033 (2000).
[36] C. Yao and E. M. Bollt, Modeling and nonlinear parameter estimation with Kronecker product representation for coupled oscillators and spatiotemporal systems, Physica D **227**, 78 (2007).
[37] O. E. Rössler, An equation for continuous chaos, Phys. Lett. A **57**, 397 (1976).
[38] M. Hénon, A two-dimensional mapping with a strange attractor, Commun. Math. Phys. **50**, 69 (1976).

[39] E. Candès, J. Romberg, and T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Info. Theory **52**, 489 (2006).

[40] E. Candès, J. Romberg, and T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Comm. Pure Appl. Math. **59**, 1207 (2006).

[41] D. Donoho, Compressed sensing, IEEE Trans. Info. Theory **52**, 1289 (2006).

[42] R. G. Baraniuk, Compressed sensing, IEEE Signal Process. Mag. **24**, 118 (2007).

[43] E. Candeès and M. Wakin, An introduction to compressive sampling, IEEE Signal Process. Mag. **25**, 21 (2008).

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

[45] Y.-C. Lai, Finding nonlinear system equations and complex network structures from data: A sparse optimization approach, Chaos **31**, 082101 (2021).

[46] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and M. A. F. Harrison, Time-series-based prediction of complex oscillator networks via compressive sensing, EPL (Europhys. Lett.) **94**, 48006 (2011).

[47] W.-X. Wang, Y.-C. Lai, C. Grebogi, and J.-P. Ye, Network reconstruction based on evolutionary-game data via compressive sensing, Phys. Rev. X **1**, 021021 (2011).

[48] R.-Q. Su, X. Ni, W.-X. Wang, and Y.-C. Lai, Forecasting synchronizability of complex networks from data, Phys. Rev. E **85**, 056220 (2012).

[49] R.-Q. Su, W.-X. Wang, and Y.-C. Lai, Detecting hidden nodes in complex networks from time series, Phys. Rev. E **85**, 065201 (2012).

[50] R.-Q. Su, Y.-C. Lai, X. Wang, and Y.-H. Do, Uncovering hidden nodes in complex networks in the presence of noise, Sci. Rep. **4**, 3944 (2014).

[51] Z. Shen, W.-X. Wang, Y. Fan, Z. Di, and Y.-C. Lai, Reconstructing propagation networks with natural diversity and identifying hidden sources, Nat. Commun. **5**, 4323 (2014).

[52] M. Scheffer, *Ecology of Shallow Lakes* (Springer Science & Business Media, 2004).

[53] M. Scheffer, Complex systems: foreseeing tipping points, Nature **467**, 411 (2010).

[54] D. B. Wysham and A. Hastings, Regime shifts in ecological systems can occur with no warning, Ecol. Lett. **13**, 464 (2010).

[55] J. M. Tylianakis and C. Coux, Tipping points in ecological networks, Trends. Plant. Sci. **19**, 281 (2014).

[56] J. Jiang, Z.-G. Huang, T. P. Seager, W. Lin, C. Grebogi, A. Hastings, and Y.-C. Lai, Predicting tipping points in mutualistic networks through dimension reduction, Proc. Nat. Acad. Sci. (USA) **115**, E639 (2018).

[57] B. Yang, M. Li, W. Tang, W. Liu, S. Zhang, L. Chen, and J. Xia, Dynamic network biomarker indicates pulmonary metastasis at the tipping point of hepatocellular carcinoma, Nat. Commun. **9**, 678 (2018).

[58] J. Jiang, A. Hastings, and Y.-C. Lai, Harnessing tipping points in complex ecological networks, J. R. Soc. Interface **16**, 20190345 (2019).

[59] Y. Meng, Y.-C. Lai, and C. Grebogi, Tipping point and noise-induced transients in ecological networks, J. R. Soc. Interface. **17**, 20200645 (2020).

[60] Y. Meng, Y.-C. Lai, and C. Grebogi, The fundamental benefits of multiplexity in ecological networks, J. R. Soc. Interface **19**, 20220438 (2022).

[61] M. Scheffer, J. Bascompte, W. A. Brock, V. Brovkin, S. R. Carpenter, V. Dakos, H. Held, E. H. Van Nes, M. Rietkerk, and G. Sugihara, Early-warning signals for critical transitions, Nature **461**, 53 (2009).

[62] J. M. Drake and B. D. Griffen, Early warning signals of extinction in deteriorating environments, Nature **467**, 456 (2010).

[63] C. Boettiger and A. Hastings, Quantifying limits to detection of early warning for critical transitions, J. R. Soc. Interface **9**, 2527 (2012).

[64] L. Chen, R. Liu, Z.-P. Liu, M. Li, and K. Aihara, Detecting early-warning signals for sudden deterioration of complex diseases by dynamical network biomarkers, Sci. Rep. **2**, 342 (2012).

[65] L. Dai, D. Vorselen, K. S. Korolev, and J. Gore, Generic indicators for loss of resilience before a tipping point leading to population collapse, Science **336**, 1175 (2012).

[66] C. Boettiger, N. Ross, and A. Hastings, Early warning signals: the charted and uncharted territories, Theor. Ecol. **6**, 255 (2013).

[67] I. A. van de Leemput, M. Wichers, A. O. Cramer, D. Borsboom, F. Tuerlinckx, P. Kuppens, E. H. van Nes, W. Viechtbauer, E. J. Giltay, S. H. Aggen, *et al.*, Critical slowing down as early warning for the onset and termination of depression, Proc. Natl. Acad. Sci. (USA) **111**, 87 (2014).

[68] N. Boers, Early-warning signals for dansgaard-oeschger events in a high-resolution ice core record, Nat. Commun. **9**, 1 (2018).

[69] Y. Zhang, P. Tino, A. Leonardis, and K. Tang, A survey on neural network interpretability, IEEE Trans. Emerg. Top. Comput. Intell. **5**, 726–742 (2021).

[70] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. **1**, 206–215 (2019).

[71] O. Alao, P. Y. Lu, and M. Soljacic, Discovering dynamical parameters by interpreting echo state networks, in *NeurIPS 2021 AI for Science Workshop* (2021).

[72] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Adaptive Computation and Machine Learning series (MIT Press, 2016).

[73] D. P. Kingma and M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[74] D. J. Rezende, S. Mohamed, and D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in *International Conference on Machine Learning* (PMLR, 2014) pp. 1278–1286.

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

[76] 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, 2006) pp. 366–381.

[77] K. McCann and P. Yodzis, Nonlinear dynamics and population disappearances, Am. Nat. **144**, 873–879 (1994).

[78] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, Algorithms for hyper-parameter optimization, in *Advances in Neural Information Processing Systems*, Vol. 24, edited by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger (Curran Associates, Inc., 2011).

[79] J. Bergstra and Y. Bengio, Random search for hyper-parameter optimization., J. Mach. Learn. Res. **13** (2012).

[80] B. Zoph and Q. V. Le, Neural architecture search with reinforcement learning, arXiv preprint arXiv:1611.01578 (2016).
[81] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, Vol. 2 (MIT press Cambridge, MA, 2006).
[82] L. Breiman, Random forests, Mach. Learn. **45**, 5 (2001).