RESEARCH ARTICLE | MARCH 03 2023

# Leveraging neural differential equations and adaptive delayed feedback to detect unstable periodic orbits based on irregularly sampled time series

Qunxi Zhu ; Xin Li ; Wei Lin



*Chaos* 33, 031101 (2023)
https://doi.org/10.1063/5.0143839

Chaos ARTICLE scitation.org/journal/cha

# Leveraging neural differential equations and adaptive delayed feedback to detect unstable periodic orbits based on irregularly sampled time series

Cite as: Chaos **33**, 031101 (2023); doi: 10.1063/5.0143839
Submitted: 26 January 2023 · Accepted: 14 February 2023 ·
Published Online: 3 March 2023

View Online Export Citation CrossMark

Qunxi Zhu,[1,2,a)] Xin Li,[3] and Wei Lin[1,2,4,5,b)]

## AFFILIATIONS

[1] Research Institute of Intelligent Complex Systems, Fudan University, Shanghai 200433, China
[2] MOE Frontiers Center for Brain Science and State Key Laboratory of Medical Neurobiology, Fudan University, Shanghai 200032, China
[3] College of Science, National University of Defense Technology, Changsha, Hunan 410073, China
[4] School of Mathematical Sciences, SCMS, SCAM, and CCSB, Fudan University, Shanghai 200433, China
[5] Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China

[a)] **Author to whom correspondence should be addressed:** qxzhu16@fudan.edu.cn
[b)] **Electronic mail:** wlin@fudan.edu.cn

## ABSTRACT

Detecting unstable periodic orbits (UPOs) based solely on time series is an essential data-driven problem, attracting a great deal of attention and arousing numerous efforts, in nonlinear sciences. Previous efforts and their developed algorithms, though falling into a category of model-free methodology, dealt with the time series mostly with a regular sampling rate. Here, we develop a data-driven and model-free framework for detecting UPOs in chaotic systems using the irregularly sampled time series. This framework articulates the neural differential equations (NDEs), a recently developed and powerful machine learning technique, with the adaptive delayed feedback (ADF) technique. Since the NDEs own the exceptional capability of accurate reconstruction of chaotic systems based on the observational time series with irregular sampling rates, UPOs detection in this scenario could be enhanced by an integration of the NDEs and the ADF technique. We demonstrate the effectiveness of the articulated framework on representative examples.



**With massive generation of datasets from real-world complex systems, model-free algorithms have been recently and overwhelmingly developed to cope with the data and solve the data-driven problems (e.g., detection of hidden and unstable periodic orbits) in various areas. These complex data are collected experimentally over time from hidden complex dynamical systems, whose explicit models are often partially or completely unknown. Although the existing methods perform well in delineating the skeleton of such complex data and the corresponding systems based on the collected data with regular sampling rates, anticipated still is the method of efficacy to deal with a more realistic case where the sampling rates of the collected data are not always regular. Fortunately, some advanced machine learning methods, including the neural differential equations (NDEs), have the capability in dealing with such irregularly sampled data. Thus, we establish a practical framework, integrating the NDEs delicately with the adaptive delayed feedback technique to pinpoint unobservable and unstable periodic orbits based only on the irregularly sampled chaotic time series. This work could be regarded as a solid step that uses the machine learning techniques to conquer the difficulties arising in the data-driven research of complex dynamical systems.**

## I. INTRODUCTION

An efficient way to understand a chaotic system is an accurate description of unstable periodic orbits (UPOs) embedded in its chaotic but invariant attractor.[1] For example, basic ergodic properties of chaotic attractors, such as fractal dimensions, Lyapunov exponents, and topological entropy, can be expressed in terms of UPOs.[2,3] UPOs are also critical to many other fields of science and engineering. A particularly important task is controlling chaos, contributed seminally by Ott, Grebogi, and Yorke (OGY),[4] where they demonstrated that a small external force of a chaotic system can be used to stabilize the UPOs embedded in the underlying chaotic attractor. In the last three decades, many interesting works, following the OGY approach, have been developed for achieving the UPO detection in various physical, biological, and ecological systems.[5–12] For example, Pyragas proposed an efficient method in Ref. 5, where the UPOs can be stabilized either by a specially designed external periodic oscillator or by the delayed self-controlling feedback without using any external force. Therefore, it is not surprising that the study of UPOs has played a vital role in the research of nonlinear complex dynamics.

In practice, the most access one to real-world nonlinear systems is the experimentally observed time series data; however, preknowledge of the explicit forms of these systems is few or even none. For example, the biophysical signals of brains can be measured via the advanced equipment, but the underlying nonlinear dynamics of brains are of still ambiguity. It, therefore, requires efficient methods for finding the hidden but principal dynamics, including those unstable periodic orbits solely from the observational data. Many works on this topic have been extensively studied.[13–17] To name a few, So *et al.*[14,15] developed a statistical method to detect UPOs from a chaotic time series by utilizing the linear dynamics around a UPO to produce a statistical index, and Dhamala *et al.*[16] addressed the detection of UPOs from experimentally measured transient chaotic time series by examining the recurrence times of trajectories in the reconstructed vector space. Although these methods are very useful in applications, they are sometimes limited to the insufficient information of linear dynamics around the relevant UPOs, the systems of discrete time or/and low-dimension, or to address the time series with uniform-interval time points sampled from continuous dynamics. Thus, it is highly anticipated for a data-driven and model-free method for accurately detecting UPOs embedded in the unknown continuous-time or/and high-dimensional (even infinite-dimensional) chaotic systems solely from the observational, irregularly sampled time series.

In recent years, machine learning, including the convolutional neural network[18] and transformers,[19] are widely applied in a variety of tasks, such as image recognition,[20] playing Go,[21] protein structure prediction,[22–24] discovering faster matrix multiplication algorithms,[25] and learning control policies.[26,27] In addition to these successful applications, recurrent neural networks, including the reservoir computing (RC),[28–31] can be used for accurately reconstructing or predicting the dynamics of complex systems from the observational time series. Typically, the RC is driven by the available time series with regular time points and maps the data space to the feature space, and then it is followed by the only trainable readout layer, mapping it back to the original data space. Notably, the RC often requires the observational time series possessing uniform-interval time points, which in turn results in the reconstruction or prediction only at discrete regular time points, not any continuous time point.

In what follows, a data-driven and model-free method, combining the neural differential equations (NDEs)[32,33] and the adaptive delayed feedback (ADF) technique, is suggested for locating the UPOs embedded in chaotic systems solely from the irregularly sampled time series. In fact, the idea is adapted partly from the previous work,[34] where the RC is used to reconstruct the underlying chaotic system from observational time series having uniform-interval time points. The framework of the RC may deal with the irregularly sampled data by using the interpolation methods, such as the natural cubic spline, to interpolate the data forming uniform-interval points. However, the recurrent computation of the RC produces an iteration system, which likely brings a large deviation when the unsampled but interpolated points are used. Instead, we in this article adopt the framework of the NDEs, which directly parameterizes the vector field of ordinary/delay differential equations (ODEs/DDEs) using neural networks and can be trained from the irregularly sampled time series. Unlike the RC, the NDEs (continuously defined dynamics) naturally obtain the data that attain arbitrary time accuracy in the training data (interpolation) and in the testing data (extrapolation) as well. Using these advantages of the NDEs, we can accurately reconstruct the underlying chaotic dynamics, and then, we employ the ADF method[8–10] to the reconstructed systems for detecting the UPOs in several representative chaotic systems even with time delays.

The rest of the article is divided into three parts. Section II briefly introduces the NDE framework for reconstructing chaotic dynamics and the ADF technique for UPO detection based on the reconstructed systems. Several representative examples are provided in Sec. III to demonstrate the effectiveness of the proposed method. Subsequently, some concluding remarks and perspectives are made in Sec. IV.

## II. METHOD: RECONSTRUCTION AND CONTROL MANIPULATION

### A. Model formulation

Consider a general nonlinear dynamical system that is modeled by a set of ordinary differential equations (ODEs),

$$\dot{\boldsymbol{x}} = \boldsymbol{F}[\boldsymbol{x}(t)], \tag{1}$$

or a set of delay differential equations (DDEs),

$$\dot{\boldsymbol{x}} = \boldsymbol{F}[\boldsymbol{x}(t-\tau), \boldsymbol{x}(t)], \tag{2}$$

where $\boldsymbol{x}(t) = [x_1(t), \ldots, x_N(t)]^\top \in \mathbb{R}^N$ is the state variable of $N$-dimension, $\boldsymbol{F} : \mathbb{R}^N$ (resp., $\mathbb{R}^N \times \mathbb{R}^N$) $\to \mathbb{R}^N$ represents a continuous vector function of the ODEs (1) [resp., the DDEs (2)], and $\tau$ is the time delay. We always assume that there is a chaotic attractor $\mathcal{A}$ generated by the system in the phase space. In this article, our goal is to detect the UPOs embedded in the chaotic attractor $\mathcal{A}$ of the system (1) or (2) without knowing its specific form but with the observational and irregularly sampled time series. Moreover, for the DDEs (2), it is assumed that the time delay $\tau$ has been estimated prior to the system reconstruction.

## B. Reconstruction of chaotic systems via NDEs

We estimate the vector field $\boldsymbol{F}$ by using the NDE framework, including the neural ODEs (NODEs)[32] and neural DDEs (NDDEs),[33] which can naturally deal with the irregularly sampled time series. In the following, we briefly introduce the NDE framework for the reconstruction of chaotic systems, and a schematic illustration of the NDE framework is depicted in Fig. 1.

For the ODEs (1), NODEs parameterize the vector field $\boldsymbol{F}$ by a neural network, i.e.,

$$\dot{\boldsymbol{z}} = \boldsymbol{f}[\boldsymbol{z}(t);\boldsymbol{\theta}], \tag{3}$$

where $\boldsymbol{\theta}$ is the vector of parameters to be trained under a predefined loss. Specifically, given the irregular observation times $t_0 = 0, t_1, \ldots, t_l = T$ and an initial state $\boldsymbol{z}(0) = \boldsymbol{x}(0)$, an ODE solver produces $\boldsymbol{z}(t_1), \ldots, \boldsymbol{z}(t_l)$, which describe the predicted state at each observation by NODEs, i.e.,

$$\boldsymbol{z}(t_i) = \boldsymbol{z}(t_{i-1}) + \int_{t_{i-1}}^{t_i} \boldsymbol{f}[\boldsymbol{z}(t);\boldsymbol{\theta}]\mathrm{d}t, i = 1, 2, \ldots, l.$$

It should be noted that modern ODE solvers can effectively obtain these states through adaptive computations.[32,35–37] With the target states at the observation times, we define the loss function as follows:

$$\mathcal{L}(\boldsymbol{z}(t_1), \ldots, \boldsymbol{z}(t_l)) = \sum_{i=1}^{l} [\boldsymbol{z}(t_i) - \boldsymbol{x}(t_i)]^2.$$

To optimize the loss function $\mathcal{L}$, we should first compute the gradients with respect to the vector $\boldsymbol{\theta}$. Actually, this can be done by using the adjoint sensitivity method.[32,38] Then, we can utilize an advanced stochastic gradient descent algorithm to train the NODEs (3). Notably, the NODEs are memory efficient, not requiring storing any intermediate quantities of the forward pass through the ODE solver and, therefore, allowing us to train the NODEs (3) with constant memory cost as a function of the number of adaptive time steps, which also allows the user to explicitly trade of the numerical precision for speed. Additionally, due to the universal approximating capability of neural networks, the NODEs (3) are, in principle, able to accurately reconstruct the underlying chaotic system (1) solely from the irregularly sampled time series.

For the DDEs (2), NDDEs define the following parameterized DDEs:

$$\dot{\boldsymbol{z}} = \boldsymbol{f}[\boldsymbol{z}(t-\tau), \boldsymbol{z}(t);\boldsymbol{\theta}]. \tag{4}$$

Different from the NODEs (3), NDDEs (4) require an initial function, $\boldsymbol{z}(t) = \phi(t)$, $t \in [-\tau, 0]$, not the initial state $\boldsymbol{z}(0) = \boldsymbol{x}(0)$ in the NODEs (3). In practice, we cannot directly obtain the initial function $\phi(t)$ but with the irregularly sampled initial history,

$$\boldsymbol{h} = \{[t_{-m}, \boldsymbol{x}(t_{-m})], \ldots, [t_{-1}, \boldsymbol{x}(t_{-1})], [t_0, \boldsymbol{x}(t_0)]\},$$

with $t_{-m} = -\tau$ and $t_0 = 0$. Hence, we need to approximate $\phi$ by a smoothed interpolation. For this purpose, the natural cubic spline is used; i.e.,

$$\boldsymbol{\phi}(t) = \mathbf{Spline}(\boldsymbol{h}, t), t \in [-\tau, 0].$$

One reason for the choice of the natural cubic spline is that $\phi(t)$ is twice continuously differentiable, a basic condition for using adaptive step-size solvers. Moreover, for accelerating the training progress of NDDEs (4), we consider the irregularly sampled time series within a time delay; i.e., $\{[t_0, \boldsymbol{x}(t_0)], [t_1, \boldsymbol{x}(t_1)], \ldots, [t_l, \boldsymbol{x}(t_l)]\}$ and $t_l \leq \tau$. Under this setup, an ODE solver can be employed to produce the predicted states,

$$\boldsymbol{z}(t_i) = \boldsymbol{z}(t_{i-1}) + \int_{t_{i-1}}^{t_i} \boldsymbol{f}[\boldsymbol{\phi}(t-\tau), \boldsymbol{z}(t);\boldsymbol{\theta}]\mathrm{d}t, i = 1, 2, \ldots, l.$$

The rest of the training process, including the loss function, is the same as the one of NODEs.

In the previous work,[34] it shows that the RC framework is able to reconstruct chaotic systems as well. Unfortunately, the classical RC can only handle the time series with regular time points and predict the states at discrete time steps, i.e., $\boldsymbol{z}(t), t = 0, \Delta t, \ldots, l\Delta t$ with the constant time increment $\Delta t$ and the number of time steps $l$.

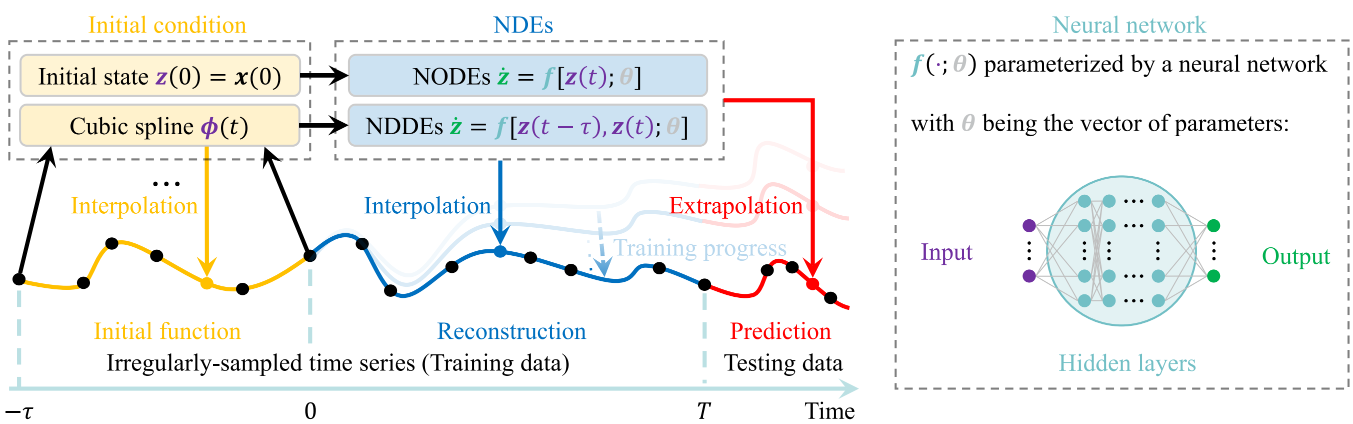


**FIG. 1.** A schematic illustration of the NDE framework, including the NODEs and NDDEs, respectively.

TABLE I. The hyperparameters of training NDEs and sampling schema that are used in our numerical examples.

| | Optimizer | Activation | Epoch size | Batch size | Hidden size/layer | Learning rate | Weight decay | *ptp* | *rsa* | $\Delta t$ | $T$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Lorenz63 | Adam | tanh | 128 | 32 | 50/2 | 0.001 | $1 \times 10^{-5}$ | 5 | 0.8 | 0.1 | 1000 |
| $x_2(t)$ (Lorenz63) | Adam | tanh | 128 | 32 | 30/2 | 0.001 | $1 \times 10^{-5}$ | 10 | 0.8 | 0.02 | 1000 |
| Mackey–Glass | Adam | tanh | 128 | 32 | 30/2 | 0.001 | $1 \times 10^{-5}$ | 5 | 0.8 | 0.2 | 1270 |

Moreover, to accurately reconstruct the vector field $\boldsymbol{F}$, Zhu *et al.*[34] approximate it by the limitation of the following equation:

$$\frac{\mathrm{d}z(t)}{\mathrm{d}t} = \lim_{\Delta t \to 0} \frac{z(t+\Delta t) - z(t)}{\Delta t} \approx \boldsymbol{F}[z(t)].$$

Accordingly, a good approximation of $\boldsymbol{F}$ requires a sufficiently small time increment $\Delta t = 0.001$ as suggested by Zhu *et al.*,[34] which results in the densely sampled time series, and, hence, increases computational costs significantly, a major bottleneck in training neural networks. However, the NDEs are of continuously defined dynamics and own the ability to deal with sparsely and irregularly sampled time series. For example, we use a low sampling rate or equivalently a large constant time increment $\Delta t = 0.1$ for the UPO detection in the Lorenz system (see Sec. III and Table I). Additionally, NDEs can obtain the state at any time point in the training data (interpolation) and the testing data (extrapolation) through a numerical solver. We also note that the RC maps the input space to high-dimensional feature space, implying that the RC may suffer from the curse of dimensionality and then requires high computational costs, as well as many parameters of the RC architecture.[34] On the contrary, NDEs directly approximate the vector field $\boldsymbol{F}$, resulting in a lightweight model.



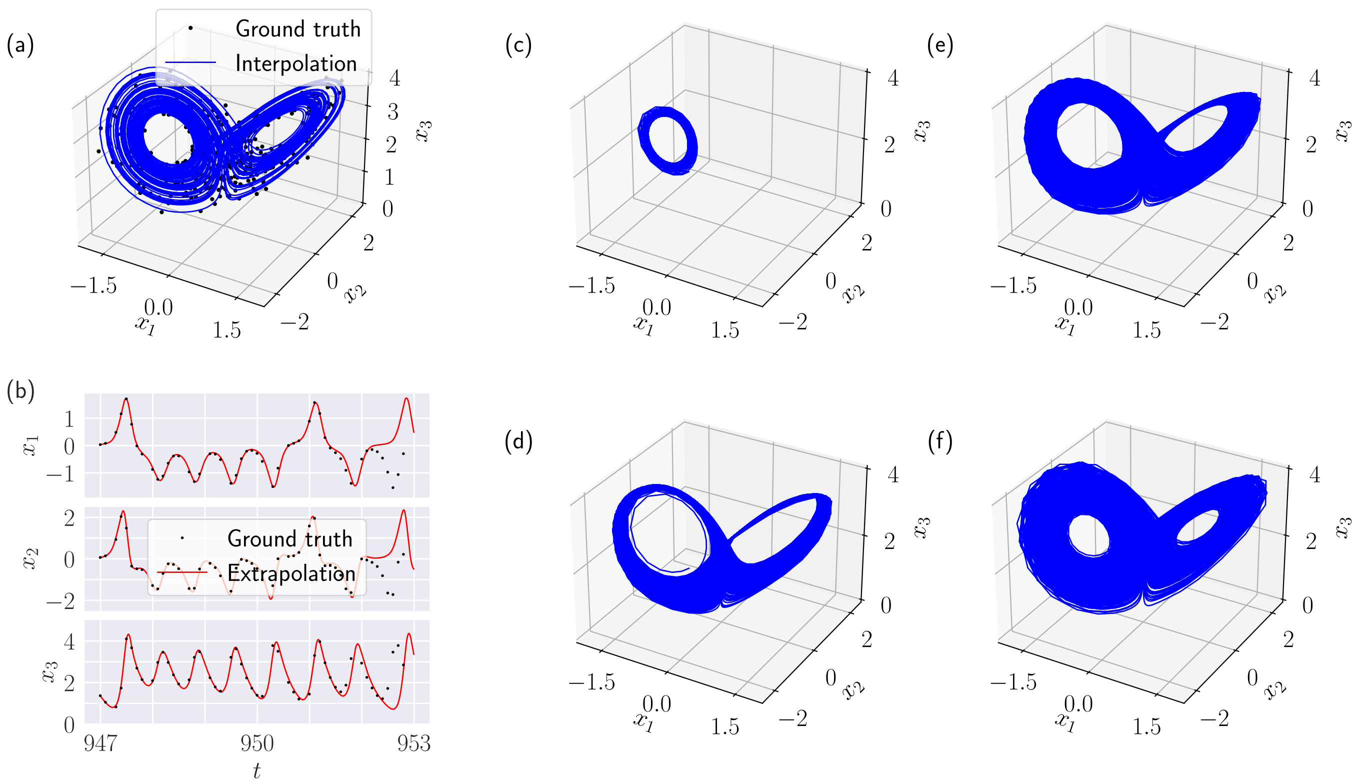


FIG. 2. Dynamics reconstruction and prediction of the Lorenz system (7) by using the NODEs (3) based solely on the irregularly sampled time series. (a) Dynamics reconstruction (blue solid line) from $t = 253$ to $t = 317$ and the irregularly sample time series (black dots) in the training phase. (b) Dynamics prediction (red solid line) from $t = 947$ to $t = 953$ and the irregularly sample time series (black dots) in the testing phase. (c)–(f) Dynamics reconstructions of the Lorenz system (7) in different training epochs, i.e., 1, 2, 14, and 80, respectively.

### C. Applying the ADF technique to the trained NDEs

Based on the approximated vector field $\boldsymbol{f}$, we focus on detecting its UPOs embedded in the chaotic attractor $\mathcal{A}$ by using the ADF technique,[8,9,34,39] which does not require any preknowledge of the UPOs. Specifically, for the case of ODEs (1), we introduce an ADF term to the approximated system (3), resulting in a controlled system as

$$\dot{\boldsymbol{u}} = \boldsymbol{f}[\boldsymbol{u}(t); \boldsymbol{\theta}] + \boldsymbol{C}(t), \tag{5}$$

where $\boldsymbol{C}(t) = \Gamma(t)\{\boldsymbol{u}[t - p(t)] - \boldsymbol{u}(t)\}$ is a delayed feedback control term. For convenience, we only control one of the components of the state; i.e., $\Gamma(t) = \text{diag}\{0, \ldots, 0, \gamma_i(t), 0, \ldots, 0\}$. This choice works well in our experiments. It should be noted that to ensure the boundedness of the controlled system and noninvasiveness of the ADF,[8,9,34,39] the time-variant delay $p(t)$ and the control gain $\gamma_i(t)$ are adaptively updated by the following DDEs:

$$\begin{aligned} \dot{p} &= -r_1 \left\{ u_i[t - p(t)] - u_i(t) \right\}, \\ \dot{\gamma}_i &= r_2 \left\{ u_i[t - p(t)] - u_i(t) \right\}^2, \end{aligned} \tag{6}$$

with $r_1$ and $r_2$ being the positive constants to be adjusted to achieve a good convergence rate.

The initial states for the controlled system (5) and the adaptive rules (6) are both taken as constants on the initial time interval. Additionally, to ensure the positiveness of the time-variant delay $p(t)$ in practice, the value of $p(t)$ is reset to a small and positive value whenever it becomes zero or exceeds a maximal threshold. As can be seen from the adaptive manners (6) that the farther the dynamic is from the expected UPOs, the faster the two variables can be adjusted for realizing stabilization. Moreover, the monotonicity of $\gamma_i(t)$ plays an important role in stabilizing the UPOs, providing a persistent and sufficiently large control gain whenever the controlled trajectory is far away from the UPOs. As long as the stabilization is achieved, the noninvasiveness of the ADF technique implies the convergence of the time-variant delay $p(t)$, corresponding to one of the unknown UPOs. For the case of DDEs (2), we can follow the same way to detect UPOs based on the trained NDDEs (4) as well.[8,9,34,39]

## III. NUMERICAL EXAMPLES

The Lorenz system and the Mackey–Glass system are taken as examples, common benchmarks for testing the methods of UPO detection, for illustrating the effectiveness of the proposed data-driven and the model-free method in Sec. II. In addition, we successfully locate the UPOs solely from a scalar time series of the Lorenz system by using the Takens delay-coordinate-embedding method.[40] In our experiments, to generate irregular time series, we randomly sample data from the regularly sampled time series with a constant time increment $\Delta t$ within the time interval $[0, T]$. Denote by *rsr* the random sampling rate from the regularly sampled time series. Other irregular sampling schemas could be considered as well. In this work, we are mainly to illustrate that our framework can deal with the irregularly sampled time series. Additionally, we choose 95% irregularly sampled time series as the training data and the

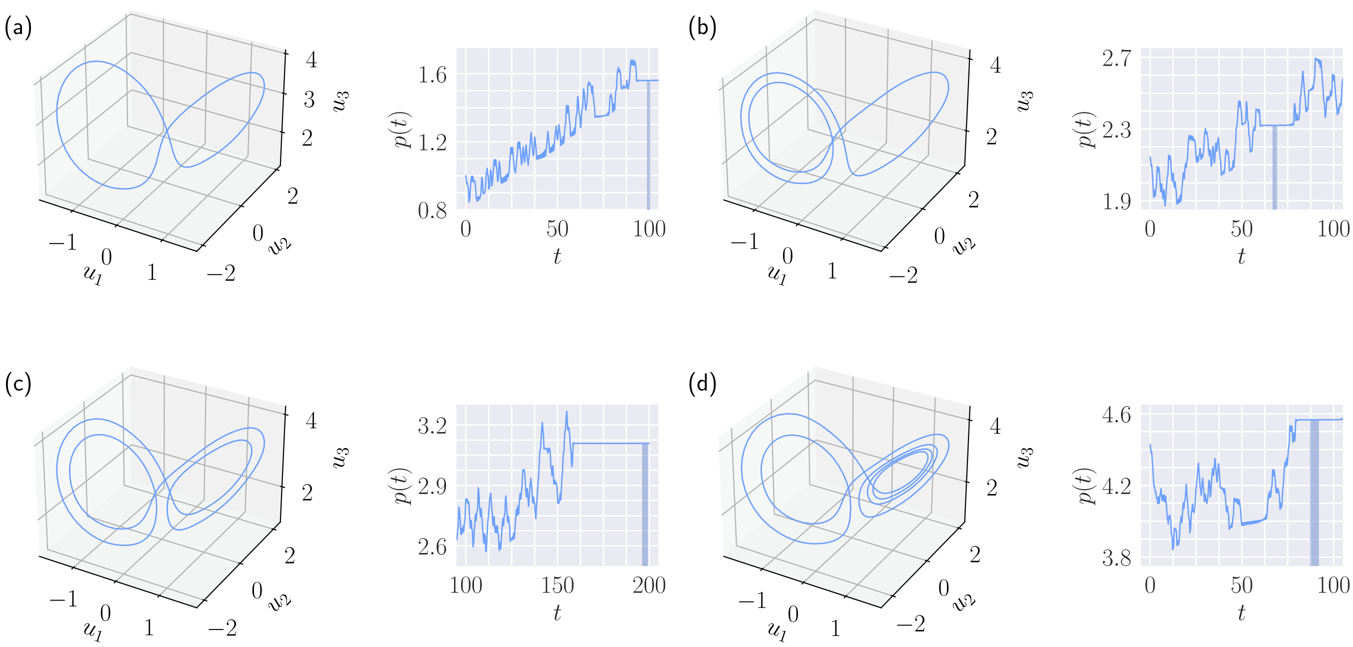


**FIG. 3.** The detected UPOs by the ADF technique (5) and (6) based on the trained NODEs (3). Four detected UPOs are shown in (a)–(d) with the periods, 1.5608, 2.3202, 3.1084, and 4.5661, respectively, which are computed by the mean value of $p(t)$ on an interval with the minimum variance (marked by a shaded interval in the right panel of each subfigure).

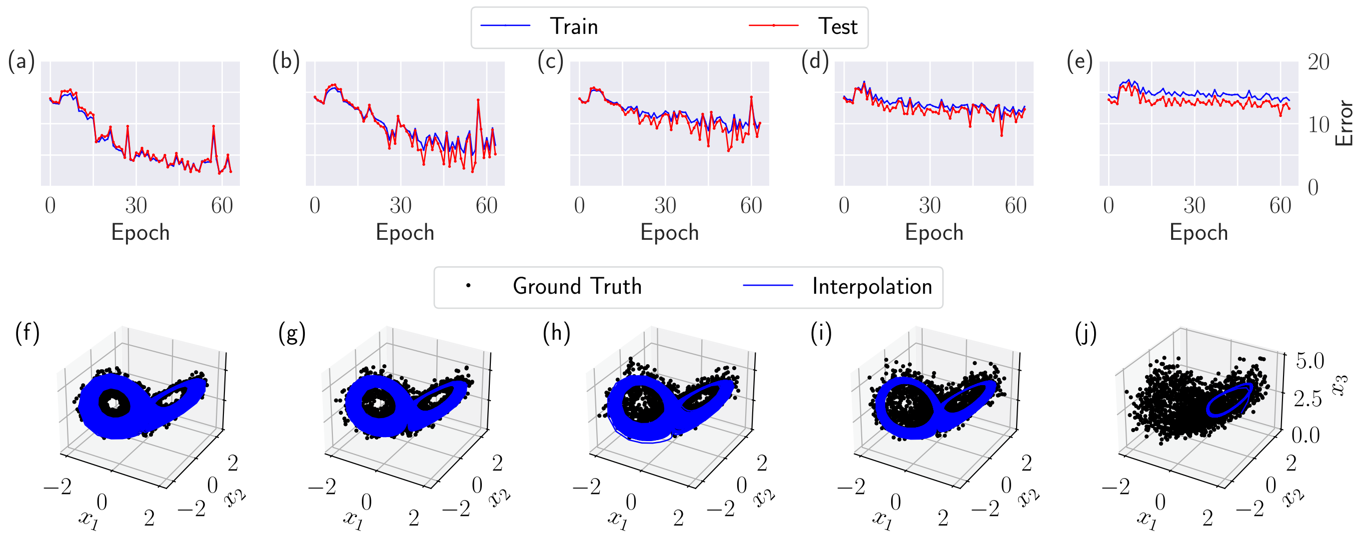


**FIG. 4.** The training and the testing errors (a)–(e) and the dynamics reconstructions (f)–(j) of the trained models under different noise levels. The variance of the noise is taken in turn from the values 0%, 0.2%, 0.5%, 1%, and 2%.

remaining as the testing data. To effectively train the model, we select the prediction time points as a small number, denoted by *ptp*. The specific parameters above, including other hyperparameters of training NDEs, are listed in Table I. The training algorithms are implemented in PyTorch, where the ODE solver, developed by Chen *et al*,[32] with the default **dopri5** method with adaptive step size during training and testing phases are employed to numerically integrate the considered nonlinear systems. In the UPO detection by using the ADF technique based on the trained NDEs, $r_1$ and $r_2$, as well as the initial values $\boldsymbol{u}(0)$, $p(0)$, and $\gamma(0)$ of the controlled system (8), are regarded as the adjusted hyperparameters. Numerically, the periods of these UPOs are determined by the mean value of the time-variant delay $p(t)$ on an interval with a low variance of $p(t)$, as well as the low differences of $p(t)$ and $\boldsymbol{u}(t)$ at the beginning and ending points. We implement the algorithm of the UPO detection through the **ddesd** solver in MATLAB.

## A. UPO detection in the Lorenz system

Here, the Lorenz system[41,42] is taken as a valuable benchmark for testing our method of stabilizing UPOs,

$$\begin{aligned}\dot{x}_1 &= 10(x_2 - x_1),\\ \dot{x}_2 &= -10x_1x_3 + 28x_1 - x_2,\\ \dot{x}_3 &= 10x_1x_2 - 8x_3/3,\end{aligned} \tag{7}$$

which contains an infinite number of UPOs embedded in its attractor.[43] According to our method, we first use the NODEs (3) to reconstruct the dynamics of the Lorenz system (7) from the irregularly sampled time series. We show the irregularly sampled time series within a given time interval and the corresponding dynamics reconstruction by the trained NODEs (3) in Fig. 2(a). Additionally, the prediction on the testing data is displayed in Fig. 2(b). The dynamics reconstructions of the Lorenz system (7) in different training epochs are shown in Figs. 2(c)–2(f). As can be seen in Fig. 2, the trained NODEs (3) have both good interpolation and extrapolation abilities in this example.

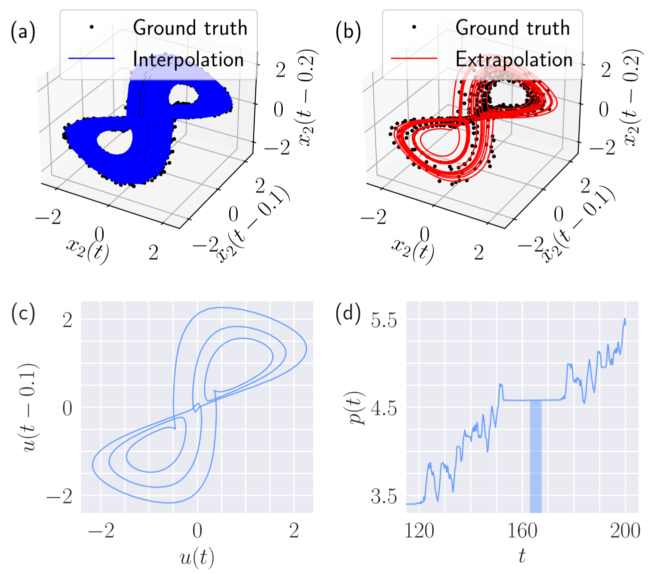


**FIG. 5.** Dynamics reconstruction (a) and prediction (b) of $x_2(t)$ of the Lorenz system (7) in the three-dimensional phase space based on the trained NODEs (3) and a detected UPO (c) by using the ADF technique (5) and (6), whose period (d) (shaded interval) is determined by the convergence of time-variant delay $p(t)$.

We apply the ADF method to the trained NODEs (3) with regard to the Lorenz system (7) to detect the UPOs, yielding a controlled system coupled with the adaptive rules (6),

$$\begin{aligned}\dot{u}_1 &= f_1(u_1, u_2, u_3; \boldsymbol{\theta}),\\ \dot{u}_2 &= f_2(u_1, u_2, u_3; \boldsymbol{\theta}) + G(t),\\ \dot{u}_3 &= f_3(u_1, u_2, u_3; \boldsymbol{\theta}),\\ \dot{p} &= -r_1\left\{u_2[t-p(t)] - u_2(t)\right\},\\ \dot{\gamma} &= r_2\left\{u_2[t-p(t)] - u_2(t)\right\}^2.\end{aligned} \tag{8}$$

To ensure the small perturbation of $C(t)$ as well as the boundedness of trajectories in the controlled time duration, the specific form $C(t)$ is suggested by Refs. 17, 34, and 39, of the following form:

$$C(t) = \mathcal{I}_{\{|G(t)|<C_0\}}G(t) + C_0\mathcal{I}_{\{|G(t)|>C_0\}} - C_0\mathcal{I}_{\{|G(t)|<-C_0\}},$$

where $C(t) = \gamma(t)\left\{u_2[t-p(t)] - u_2(t)\right\}$, $\mathcal{I}_S$ is the indicator of the set $S$, and $C_0$ is a small constant that can be adjusted to control the intensity of the perturbation $G(t)$. As shown in Fig. 3, several UPOs are detected with a good consistency with the reported UPOs.[44]

Notably, an accurate dynamics reconstruction of the underlying system is a key step for achieving the UPO detection. We, therefore, assess the robustness of the dynamics reconstructions by considering the Lorenz system with the measured noise as a benchmark. In particular, we generate the time series perturbed by the multiplicative noise, i.e., $\hat{x}_i(t) = x_i(t)[1 + n(t)]$ with $i = 1, 2, 3$, where $n(t)$ is a noise term sampled from the Gaussian normal distribution with zero mean and the adjustable variance. Figures 4(a)–4(e) show the training and the testing errors under different noise levels by varying the variance. The corresponding dynamics reconstructions of the trained models are displayed in Figs. 4(f)–4(j). It can be seen that the training and the testing errors increase monotonically with the noise strength. The reconstructed systems exhibit two lobes, until the noise variance increases through a threshold, approximately 1%. Notice that the solution of ODEs is determined by the initial value. Then, if the noise increases, the learning process of dynamics reconstructions may easily fail due to the large deviation from the true initial value. This suggests that we may improve the robustness of our framework by first estimating the initial value more accurately based on the time series via curve-fitting schemes or using the neural ODE processes[45] to handle



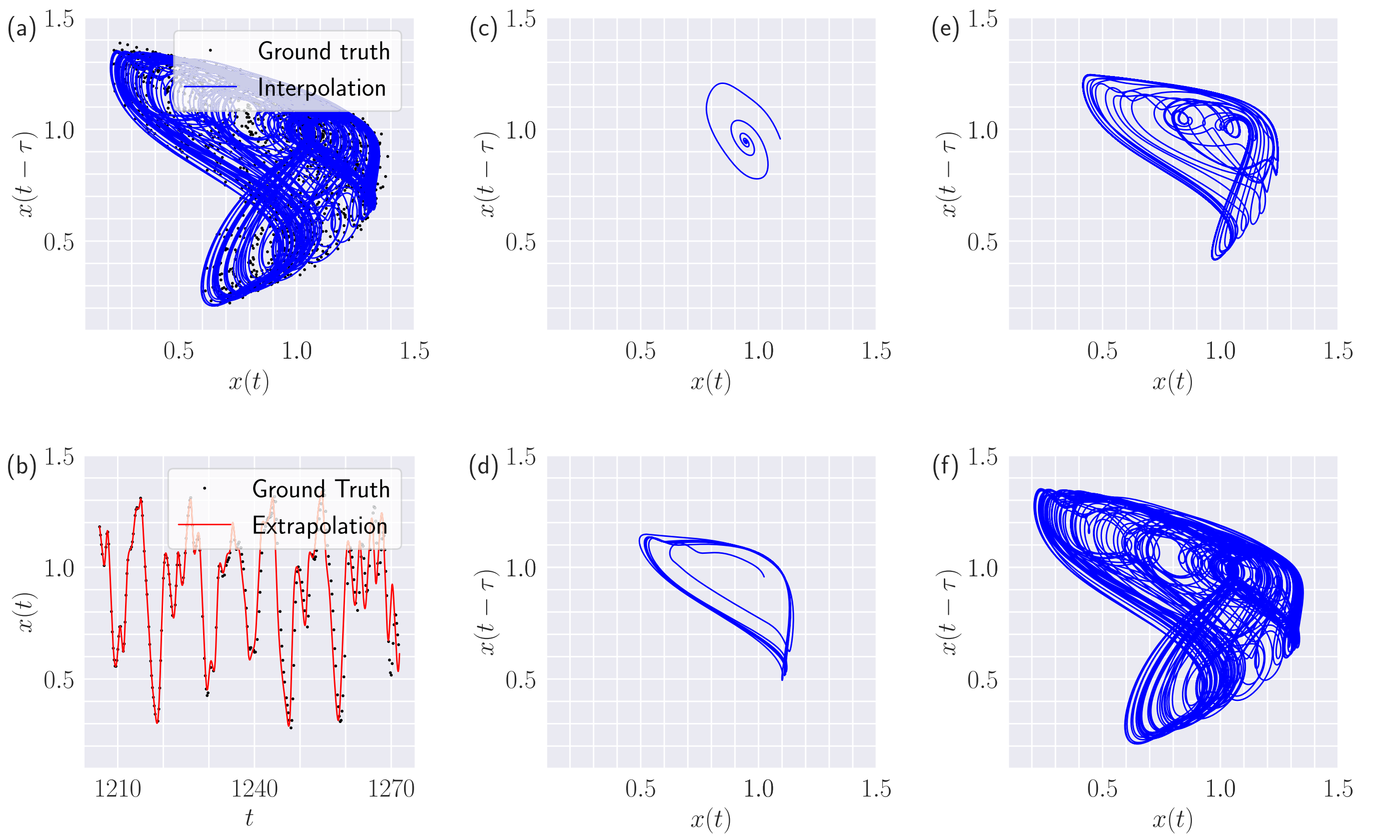


**FIG. 6.** Dynamics reconstruction and prediction of the Mackey–Glass system (10) by using the NDDEs (4) based solely on the irregularly sampled time series. (a) Dynamics reconstruction (blue solid line) from $t = 123$ to $t = 621$ and the irregularly sample time series (black dots) in the training phase. (b) Dynamics prediction (red solid line) from $t = 1206$ to $t = 1272$ and the irregularly sample time series (black dots) in the testing phase. (c)–(f) Dynamics reconstructions of the Mackey–Glass system (10) in different training epochs, i.e., 1, 7, 13 and 64, respectively.

the noise. This could be further studied as one of our future directions.

Here, we consider the case that only one state variable of the Lorenz system (7) can be observed, i.e., the irregularly sampled time series of $x_2(t)$. In this case, we use the classic Takens delay-coordinate-embedding method[40] to reconstruct the phase space from the time series, recovering unobserved degrees of freedom. Particularly, the reconstructed system is consisted of other time-delayed state variables, resulting in an augmented vector $\boldsymbol{y}(t) = [x_2(t), x_2(t-\tau), \ldots, x_2(t-(L-1)\tau)]^\top \in \mathbb{R}^L$, where $\tau = 0.1$ is a constant time delay and $L = 3$ is the embedding dimension. Here, we assume that $\boldsymbol{y}(t)$ obeys the governing equation,

$$\dot{\boldsymbol{y}} = \hat{\boldsymbol{F}}[\boldsymbol{y}(t)], \tag{9}$$

where $\boldsymbol{F} : \mathbb{R}^L \to \mathbb{R}^L$ models the vector field of this "virtual" system. In fact, the system (9) and the Lorenz system (7) are related through diffeomorphism. Thus, it is still possible to locate the UPOs solely from the scalar time series.

In practice, to obtain these time-delayed state variables $x_2(t - i\tau)$, we interpolate them by the cubic interpolation method based on the time series of $x_2(t)$. Then, the following steps are the same as the case of the fully observable Lorenz system (7). We routinely use the NODEs (3) to approximate the system (9) and then apply the ADF technique to this approximated system to locate the unknown UPOs. As shown in Fig. 5, the NDEs achieve good performances on the training data (interpolation) in Fig. 5(a) and testing data (extrapolation) in Fig. 5(b), and a UPO with an approximated period, 4.5804, is successfully detected in Fig. 5(c) according to the convergence of the time-variant delay $p(t)$ in Fig. 5(d).

### B. UPO detection in the Mackey–Glass system

Finally, we test our method of locating the UPOs in the Mackey–Glass system, a scalar time-delayed model for blood cell regeneration, regarded as an infinite-dimensional model,[46] of the following form:

$$\dot{x} = \frac{ax(t-\tau)}{1 + x^b(t-\tau)} - cx, \tag{10}$$

where $x$ describes the concentration of the circulating blood cells, $\tau$ is a constant feedback time delay, and $a$, $b$, and $c$ are the parameters of biological significance. Here, we choose the parameters $a = 2$, $b = 10$, $c = 1$, and $\tau = 3.18$, and the system exhibits chaotic dynamics. Based on the generated irregularly sampled time series, we first use the NDDEs (4) to reconstruct the dynamics of the Mackey–Glass system (10). The partial time series and the corresponding dynamics reconstruction by the trained NDDEs (4) are shown in Fig. 6(a). In addition, Fig. 6(b) shows good extrapolation ability on the testing data. As shown in Figs. 6(c)–6(f), the NDDEs (4) gradually capture the intrinsic dynamics of the Mackey–Glass system (10) along with the training procedure. Next, we directly apply the ADF technique to the trained NDDEs (4). The detected UPOs of the trained NDDEs (4), as well as the original Mackey–Glass system (10), are displayed in Fig. 7 according to the convergence of the time-variant delay $p(t)$. As can be seen, not only the shape but also the period ($p = 7.5994$) of the detected UPO from the NDDEs (4) are almost consistent with that ($p^* = 7.6001$) from the original system. It indicates that our data-driven and model-free method has the ability to detect UPOs in high-dimensional chaotic systems solely from the irregularly sampled time series.

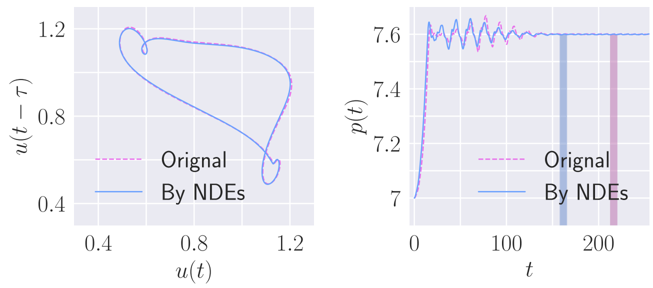


**FIG. 7.** The detected UPOs from the original Mackey–Glass system (10) and the corresponding approximated NDDEs (4) by using the ADF technique (5) and (6). As can be seen, the shape, as well as the period ($p = 7.5994$) of the detected UPO from the trained NDDEs (4), is almost consistent with that ($p^* = 7.6001$) from the original system.

## IV. CONCLUSION

In conclusion, we have articulated a data-driven and model-free method, integrating the NDEs and the ADF technique, to detect UPOs embedded in chaotic attractors solely from the irregularly sampled time series. Particularly, the NDEs successfully reconstruct the dynamical systems with/without time delay from the fully or partially observational irregularly sampled time series, based on which the ADF technique can be employed to faithfully detect the UPOs without any prior information on system *per se*. Our work suggests that combining the advanced machine learning frameworks with the classical theories and methods developed in nonlinear dynamics could shed light on solving the longstanding problems in complex systems efficiently and accurately.

## ACKNOWLEDGMENTS

Q.Z. was supported by the China Postdoctoral Science Foundation (No. 2022M720817), by the Shanghai Postdoctoral Excellence Program (No. 2021091), and by the STCSM (Nos. 21511100200 and 22ZR1407300). W.L. was supported by the National Natural Science Foundation of China (No. 11925103) and by the STCSM (Nos. 22JC1402500, 22JC1401402, and 2021SHZDZX0103).

## AUTHOR DECLARATIONS

### Conflict of Interest

The authors have no conflicts to disclose.

### Author Contributions

Q.Z. and X.L. contributed equally to this work.

**Qunxi Zhu:** Conceptualization (lead); Formal analysis (equal); Investigation (equal); Methodology (equal); Visualization (equal);

Writing – original draft (equal); Writing – review & editing (equal). **Xin Li:** Formal analysis (equal); Investigation (equal); Methodology (equal); Visualization (equal). **Wei Lin:** Conceptualization (equal); Writing – review & editing (equal).

## DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## REFERENCES


[1] E. Ott, *Chaos in Dynamical Systems* (Cambridge University Press, 2002).
[2] D. Auerbach, P. Cvitanović, J.-P. Eckmann, G. Gunaratne, and I. Procaccia, "Exploring chaotic motion through periodic orbits," Phys. Rev. Lett. **58**(23), 2387 (1987).
[3] P. Cvitanović, "Invariant measurement of strange sets in terms of cycles," Phys. Rev. Lett. **61**(24), 2729 (1988).
[4] E. Ott, C. Grebogi, and J. Yorke, "Controlling chaos," Phys. Rev. Lett. **64**(11), 1196 (1990).
[5] K. Pyragas, "Continuous control of chaos by self-controlling feedback," Phys. Lett. A **170**(6), 421–428 (1992).
[6] G. Chen and X. Yu, *Chaos Control: Theory and Applications* (Springer, Berlin, 2003).
[7] D. Huang, "Stabilizing near-nonhyperbolic chaotic systems with applications," Phys. Rev. Lett. **93**(21), 214101 (2004).
[8] W. Lin and H. Ma, "Failure of parameter identification based on adaptive synchronization techniques," Phys. Rev. E **75**(6), 066212 (2007).
[9] H. Ma and W. Lin, "Nonlinear adaptive synchronization rule for identification of a large amount of parameters in dynamical models," Phys. Lett. A **374**(2), 161–168 (2009).
[10] W. Lin and H. Ma, "Synchronization between adaptively coupled systems with discrete and distributed time-delays," IEEE Trans. Autom. Control **55**(4), 819–830 (2010).
[11] T. Ge, X. Tian, J. Kurths, J. Feng, and W. Lin, "Achieving modulated oscillations by feedback control," Phys. Rev. E **90**, 022909 (2014).
[12] W. Lin, X. Chen, and S. Zhou, "Achieving control and synchronization merely through a stochastically adaptive feedback coupling," Chaos **27**, 073110 (2017).
[13] D. Lathrop and E. Kostelich, "Characterization of an experimental strange attractor by periodic orbits," Phys. Rev. A **40**(7), 4028 (1989).
[14] P. So, S. J. Schiff, D. T. Kaplan, E. Ott, T. Sauer, and C. Grebogi, "Detecting unstable periodic orbits in experimental data," Phys. Rev. Lett. **76**, 4705 (1996).
[15] P. So, E. Ott, T. Sauer, B. J. Gluckman, C. Grebogi, and S. J. Schiff, "Extracting unstable periodic orbits from chaotic time series data," Phys. Rev. E **55**, 5398 (1997).
[16] M. Dhamala, Y.-C. Lai, and E. J. Kostelich, "Detecting unstable periodic orbits from transient chaotic time series," Phys. Rev. E **61**(6), 6485 (2000).
[17] H. Ma, W. Lin, and Y. Lai, "Detecting unstable periodic orbits in high-dimensional chaotic systems from time series: Reconstruction meeting with adaptation," Phys. Rev. E **87**(5), 050901 (2013).
[18] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks* (MIT Press, 1995), Vol. 3361, Iss. 10.
[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2017).
[20] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2012), pp. 1097–1105.
[21] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, and S. Dieleman, "Mastering the game of go with deep neural networks and tree search," Nature **529**(7587), 484 (2016).
[22] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, and A. Bridgland, "Highly accurate protein structure prediction with alphafold," Nature **596**(7873), 583–589 (2021).
[23] G. Xu, Q. Wang, and J. Ma, "OPUS-TASS: A protein backbone torsion angles and secondary structure predictor based on ensemble neural networks," Bioinformatics **36**(20), 5021–5026 (2020).
[24] G. Xu, Q. Wang, and J. Ma, "OPUS-Rota4: A gradient-based protein side-chain modeling framework assisted by deep learning-based predictors," Briefings Bioinf. **23**(1), bbab529 (2022).
[25] A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F. J. R. Ruiz, J. Schrittwieser, G. Swirszcz, and D. Silver, "Discovering faster matrix multiplication algorithms with reinforcement learning," Nature **610**(7930), 47–53 (2022).
[26] J. Zhang, Q. Zhu, and W. Lin, "Neural stochastic control," in *Advances in Neural Information Processing Systems*, edited by A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho (Curran Associates, Inc., 2022).
[27] J. Zhang, Q. Zhu, W. Yang, and W. Lin, "SYNC: Safety-aware neural control for stabilizing stochastic delay-differential equations," in *International Conference on Learning Representations* (OpenReview, 2023).
[28] 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).
[29] X. Ji, Y. Wu, W. Sheng, and W. Lin, "Identification of interactions in fractional-order systems with high dimensions," Chaos **24**, 023119 (2014).
[30] J. Pathak, Z. Lu, B. Hunt, M. Girvan, and E. Ott, "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data," Chaos **27**(12), 121102 (2017).
[31] Z. Lu, B. Hunt, and E. Ott, "Attractor reconstruction by machine learning," Chaos **28**(6), 061104 (2018).
[32] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2018), Vol. 31.
[33] Q. Zhu, Y. Guo, and W. Lin, "Neural delay differential equations," in *International Conference on Learning Representations* (OpenReview, 2021).
[34] Q. Zhu, H. Ma, and W. Lin, "Detecting unstable periodic orbits based only on time series: When adaptive delayed feedback control meets reservoir computing," Chaos **29**(9), 093125 (2019).
[35] C. Runge, "Über die numerische Auflösung von Differentialgleichungen," Math. Ann. **46**(2), 167–178 (1895).
[36] W. Kutta, "Beitrag zur naherungsweisen integration totaler Differentialgleichungen," Z. Math. Phys. **46**, 435–453 (1901).
[37] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations, 1, Nonstiff Problems* (Springer, 1993).
[38] L. S. Pontryagin, *Mathematical Theory of Optimal Processes* (CRC Press, 1987).
[39] W. Lin, H. Ma, J. Feng, and G. Chen, "Locating unstable periodic orbits: When adaptation integrates into delayed feedback control," Phys. Rev. E **82**(4), 046214 (2010).
[40] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence* (Springer, 1981), pp. 366–381.
[41] E. Lorenz, "Deterministic nonperiodic flow," J. Atmos. Sci. **20**(2), 130–141 (1963).
[42] C. Sparrow, *The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors* (Springer, New York, 1982).
[43] B. Boghosian, A. Brown, J. Lätt, H. Tang, L. Fazendeiro, and P. Coveney, "Unstable periodic orbits in the Lorenz attractor," Philos. Trans. R. Soc. A **369**(1944), 2345–2353 (2011).
[44] Z. Galias and W. Tucker, "Short periodic orbits for the Lorenz system," in *International Conference on Signals and Electronic Systems* (IEEE, Piscataway, NJ, 2008), pp. 285–288.
[45] A. Norcliffe, C. Bodnar, B. Day, J. Moss, and P. Liò, "Neural ODE processes," arXiv:2103.12413 (2021).
[46] M. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," Science **197**(4300), 287–289 (1977).