Chaos
An Interdisciplinary Journal of Nonlinear Science

RESEARCH ARTICLE | JULY 09 2024

A few-shot identification method for stochastic dynamical systems based on residual multipeaks adaptive sampling

Xiao-Kai An ; Lin Du ; Feng Jiang ; Yu-Jia Zhang; Zi-Chen Deng; Jürgen Kurths

Chaos 34, 073118 (2024)
https://doi.org/10.1063/5.0209779

A few-shot identification method for stochastic dynamical systems based on residual multipeaks adaptive sampling

Cite as: Chaos 34, 073118 (2024); doi: 10.1063/5.0209779
Submitted: 23 March 2024 · Accepted: 18 June 2024 ·
Published Online: 9 July 2024

View Online Export Citation CrossMark

Xiao-Kai An,[1,2] Lin Du,[1,2,a)] Feng Jiang,[1,3] Yu-Jia Zhang,[2] Zi-Chen Deng,[1] and Jürgen Kurths[4]

AFFILIATIONS

[1]MIIT Key Laboratory of Dynamics and Control of Complex Systems, Northwestern Polytechnical University, Xi'an 710072, China
[2]School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710072, China
[3]School of Mechanics, Civil Engineering and Architecture, Northwestern Polytechnical University, Xi'an 710072, China
[4]Potsdam Institute for Climate Impact Research, Potsdam 14473, Germany

[a)]**Author to whom correspondence should be addressed:** lindu@nwpu.edu.cn

ABSTRACT

Neural networks are popular data-driven modeling tools that come with high data collection costs. This paper proposes a residual-based multipeaks adaptive sampling (RMAS) algorithm, which can reduce the demand for a large number of samples in the identification of stochastic dynamical systems. Compared to classical residual-based sampling algorithms, the RMAS algorithm achieves higher system identification accuracy without relying on any hyperparameters. Subsequently, combining the RMAS algorithm and neural network, a few-shot identification (FSI) method for stochastic dynamical systems is proposed, which is applied to the identification of a vegetation biomass change model and the Rayleigh–Van der Pol impact vibration model. We show that the RMAS algorithm modifies residual-based sampling algorithms and, in particular, reduces the system identification error by 76% with the same sample sizes. Moreover, the surrogate model accurately predicts the first escape probability density function and the P bifurcation behavior in the systems, with the error of less than 1.59×10^{-2}. Finally, the robustness of the FSI method is validated.

Dynamical modeling is the basis for the analysis and control of complex systems. With the advancement of data science, data-driven dynamical modeling has emerged as a hot research topic. Neural network techniques have made significant progress in the field of stochastic dynamical system identification, which requires a large amount of trajectory data to estimate statistical features. Therefore, it is a challenging task to develop efficient sampling algorithms to reduce the training cost of a neural network. Classical residual-based sampling algorithms, such as residual-based adaptive refinement and residual-based adaptive distribution, mainly focus on a single maximum in the residual space, lacking the utilization of global information and being difficult to generalize due to their dependence on hyperparameters. To overcome these problems, we propose a residual-based multipeaks adaptive sampling (RMAS) algorithm that does not rely on any hyperparameters. Furthermore, this work combines the RMAS algorithm with the neural network to develop a few shot identification (FSI) method for stochastic dynamical systems. Numerical results show that, compared to classical sampling algorithms, RMAS achieves higher system identification accuracy with fewer samples.

I. INTRODUCTION

With the rapid development of data science, data-driven modeling has become a powerful alternative and complementary approach to first-principles modeling.[1–4] In the last decade, neural network-based identification has clearly become the most popular black-box modeling technique.[5] Machine learning methods including neural network have been gradually applied to inverse problems for ordinary, partial and stochastic differential equations.[6–9]

Neural networks embedded with physical information can be effectively applied to the study of dynamical systems, such as Hamiltonian neural network,[10] Lagrangian neural network,[11] physics-informed neural network,[12] and Neural ODE.[13] Despite

the good performance of these neural networks, there is still room for improvement in the design of sampling algorithms. Recently the stochastic physics-informed neural ordinary differential equation (SPINODE) method has been proposed, which can efficiently identify stochastic differential equations with Gaussian noise excitation.[14] However, SPINODE requires a large amount of trajectory data from the same initial value to describe main statistical characteristics of the system. It is challenging to develop an efficient sampling algorithm to reduce the cost of acquiring trajectory data. In practical engineering, a small number of samples means that fewer sensors can be used to avoid damage to the equipment structure and to save costs.

The residual-based adaptive refinement (RAR) algorithm is a classical and effective method in the sampling step of treating a dynamical system inverse problem.[15] In order to extract more information from the residuals, the residual-based adaptive distribution (RAD) algorithm has been proposed recently.[16] There has been a growing wealth of studies pertaining to the estimation of residual probability density functions.[17–20] However, there is a paucity of research examining the utilization of global characteristics of the sampling space for selecting suitable sampling positions. The method of dividing the sampling space into multiple subregions effectively utilizes global information of the sampling space.[21] However, such a division is inflexibly dependent on many hyperparameters. The existing sampling methods lack the consideration of multiple maxima of the residual probability density function $p(x)$. To overcome this drawback, a new residual-based multipeaks adaptive sampling (RMAS) algorithm is proposed in this paper. The method composed of a neural network and an RMAS algorithm is utilized for the basic tasks of stochastic dynamical system identification.

The RMAS algorithms and the neural network are trained alternately. After the neural network completes one epoch of training, RMAS estimates the probability density function $p(x)$ of the residual based on prior knowledge of drift and diffusion coefficients. Then, the RMAS algorithm collects a sample at each maximum in the residual probability density function to form a new sample set, which is used to support the training of the neural network in the next epoch. This process continues until the error of system identification is below a preset threshold. Compared to classical residual-based sampling algorithms, RMAS has the advantage of not relying on hyperparameters.

The paper is organized as follows: Section II describes the process of combining the RMAS algorithm with neural network to identify stochastic differential equations. Two basic examples are studied using the identification method in Sec. III: a vegetation biomass change model and the Rayleigh–Van der Pol impact vibration model. Section IV discusses the robustness of the identification method. Concluding comments are made in Sec. V.

II. METHOD

A stochastic dynamical system under parametric excitation in an n dimensional state space can be defined by as

$$\frac{\mathrm{d}}{\mathrm{d}t}X_j(t) = f_j(\boldsymbol{X}(t)) + h_j(\boldsymbol{X}(t))W_j(t), \quad j = 1, 2, \ldots, n, \tag{1}$$

where $\boldsymbol{X}(t) = [X_1(t), X_2(t), \ldots, X_n(t)]^{\mathrm{T}}$ is the state variable of an n degree of freedom system, $W_j(t)(j = 1, 2, \ldots, n)$ are Gaussian white noise, and their correlation function are

$$E\left[W_j(t)W_j(t+\tau)\right] = 2\pi K_j\delta(\tau), \quad j = 1, 2, \ldots, n. \tag{2}$$

Equation (2) can be converted into Itô form given in Eq. (3), where $B_j(t)(j = 1, 2, \ldots, n)$ are independent unit Wiener processes,

$$\mathrm{d}X_j(t) = f_j(\boldsymbol{X}(t))dt + h_j(\boldsymbol{X}(t))dB_j(t). \tag{3}$$

A. Data collection

Multiple trajectory data from the same initial values are used to extract statistical features (for example, moment information). To achieve accurate moment information, we use the Euler–Milstein method [Eq. (4)] as a numerical simulation method to generate the training data, which is a discretized form of t,

$$X_j(t + \Delta t) = X_j(t) + f_j(\boldsymbol{X}(t))\,\Delta t + h_j(\boldsymbol{X}(t))\Delta W_j, \tag{4}$$

where $j = 1, 2, \ldots, n$. Despite starting from the same initial values, the trajectories of numerical solutions of stochastic differential equations are different due to the diffusion term. Therefore, Eqs. (5) and (6) are used to calculate the first and second moments of the trajectory data as statistical values, where the mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and the covariance $P \in \mathbb{R}^{n\times n}$. For each initial value, the stochastic differential equation is solved N times with a step size of L,

$$\boldsymbol{\mu}(t_k) = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{X}(t_k), \tag{5}$$

$$P(t_k) = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{X}(t_k) - \boldsymbol{\mu}(t_k))(\boldsymbol{X}(t_k) - \boldsymbol{\mu}(t_k))^{\mathrm{T}}. \tag{6}$$

B. Neural network training

It is challenging to estimate the true state distribution from limited state data. Therefore, this paper utilizes matching of moment information for neural network training. Unscented transformation is a useful tool for estimating moment information by using a set of data points called sigma points.[22–24] Sigma points $z^{(i)}(t_k)$ $(i = 1, 2, \ldots, 2n)$ have dynamics in the form of differential equations, enabling the prediction of moment information, which can be calculated using Eq. (7) when the mean $\boldsymbol{\mu}(t_k)$ [Eq. (5)] and the covariance $P(t_k)$ [Eq. (6)] of trajectory data are known,

$$\begin{aligned} \boldsymbol{z^{(0)}}(t_k) &= \boldsymbol{\mu}(t_k), \\ z^{(i)}(t_k) &= \boldsymbol{\mu}(t_k) + [A]_i, i = 1, \ldots, n, \\ z^{(i)}(t_k) &= \boldsymbol{\mu}(t_k) - [A]_{i-1}, i = n+1, \ldots, 2n, \end{aligned} \tag{7}$$

where matrix $A = \sqrt{n + \lambda P}$ denotes the Cholesky decomposition and $[A]_i$ is the column i of the matrix A. The weights of the sigma points relative to the mean and covariance are calculated

using Eq. (8),

$$w_0^{(c)}(t_k) = \frac{\lambda}{(n+\lambda) - (1-\alpha^2+\beta)},$$
$$w_0^{(m)}(t_k) = \frac{\lambda}{n+\lambda}, \tag{8}$$
$$w_i^{(c)}(t_k) = W_i^{(m)}(t_k) = \frac{1}{2(n+\lambda)}, i = 1, \ldots, 2n,$$

where α, β, and λ are hyperparameters.

Neural ODE networks can approximate any continuous function, and here, they are used to a nonlinear map $F(z^{(i)};\theta)$ of sigma points, satisfying the dynamical equation $\dot{\boldsymbol{z}}^{(i)} = F(z^{(i)};\theta)$. $z^{(i)}(t_{k+1};\theta)$ can be calculated through integrating the Neural ODE when $z^{(i)}(t_k)$ is known. Then, the mean and covariance at time t_{k+1} can be predicted using Eqs. (9) and (10),

$$\hat{\boldsymbol{\mu}}(t_{k+1};\theta) = \sum_{i=0}^{2n} w_i^{(m)} z^{(i)}(t_{k+1};\theta), \tag{9}$$

$$\widehat{P}(t_{k+1};\theta) = \sum_{i=0}^{2n} w_i^{(c)} \left(z^{(i)}(t_{k+1};\theta) - \hat{\boldsymbol{\mu}}(t_{k+1};\theta)\right) \times \left(z^{(i)}(t_{k+1};\theta) - \hat{\boldsymbol{\mu}}(t_{k+1};\theta)\right)^{\mathrm{T}}. \tag{10}$$

Two independent neural networks $F(z^{(i)};\theta_1)$ and $F(z^{(i)};\theta_2)$ can be used to predict the mean and covariance, respectively, because the disturbance in Eq. (1) is Gaussian white noise with a mean of zero.[14] In particular, the following two optimization problems are solved separately during the neural network training process,

$$\theta_1^{\star} = \arg\min_{\theta_1} \sum_{k=0}^{L-1} \left\| \boldsymbol{\mu}(t_{k+1}) - \hat{\boldsymbol{\mu}}(t_{k+1}|t_k;\theta_1) \right\|^2, \tag{11}$$

$$\theta_2^{\star} = \arg\min_{\theta_2} \sum_{k=0}^{L-1} \left\| P(t_{k+1}) - \widehat{P}(t_{k+1}|t_k;\theta_1^{\star},\theta_2) \right\|^2. \tag{12}$$

C. Residual based sampling

The drift term $\Phi(\boldsymbol{X}) = [f_1(\boldsymbol{X}), f_2(\boldsymbol{X}), \ldots, f_n(\boldsymbol{X})]^{\mathrm{T}}$ and the diffusion term $\Psi(\boldsymbol{X}) = [h_1(\boldsymbol{X}), h_2(\boldsymbol{X}), \ldots, h_n(\boldsymbol{X})]^{\mathrm{T}}$ in Eq. (3) can be estimated, respectively, by $F\left(z^{(i)};\theta_1\right)$ and $F\left(z^{(i)};\theta_2\right)$ when $z^{(i)}$ is replaced with $\boldsymbol{X}$. The residual $\varepsilon_r(\boldsymbol{X})$ of this estimation can be calculated as

$$\varepsilon_r(\boldsymbol{X}) = \|F(\boldsymbol{X};\theta_1) - \Phi(\boldsymbol{X})\|_1 + \|F(\boldsymbol{X};\theta_2) - \Psi(\boldsymbol{X})\|_1. \tag{13}$$

The domains of both $\Phi(\boldsymbol{X})$ and $\Psi(\boldsymbol{X})$ are assumed to be $\Omega \subset R^n$. To calculate the residual, some prior information about the drift and the diffusion term needs to be known. The RMAS method assumes that there only a small number of discrete points $X \in \Omega_0 \subset \Omega$ that satisfy $\Phi(\boldsymbol{X})$ and $\Psi(\boldsymbol{X})$ are known. The probability density function (PDF) of the residual $p(\boldsymbol{X}), X \in \Omega$ will be obtained by fitting a polynomial to the discrete values of $\varepsilon_r(\boldsymbol{X}), X \in \Omega_0$.

In this paper, residual-based multi-peak adaptive sampling (RMAS) algorithm is proposed. Fitting $\varepsilon_r(\boldsymbol{X})$ with a polynomial of order greater than 2 is necessary to guarantee the existence of $p''(\boldsymbol{X})$.

ALGORITHM 1. RMAS algorithm without hyperparameters.

Input: PDF of the residual $p(\boldsymbol{X})$; Sample set Γ_k
Output: Sample set of the $(k+1)$th epoch Γ_{k+1}
$\omega \leftarrow \{\boldsymbol{X} | p'(\boldsymbol{X}) = 0 \wedge p''(\boldsymbol{X}) < 0 \wedge \boldsymbol{X} \in \Omega\}$
for $\boldsymbol{X}_i \in \omega_n$ **do**
  $A \leftarrow U(\boldsymbol{X}_i, 5) \cap \Omega \setminus \Gamma_k$
  $B \leftarrow \arg\max_{\boldsymbol{X}} \{p(\boldsymbol{X}) | \boldsymbol{X} \in A\}$
  $\Gamma_k \leftarrow \Gamma_k \cup B$
end for
$\Gamma_{k+1} \leftarrow \Gamma_k$

Based on the definition of the maximum point and the continuity of $p(\boldsymbol{X})$, it is easy to find the set of maximum points, which is defined as $\omega = \{\boldsymbol{X} | p'(\boldsymbol{X}) = 0 \text{ and } p''(\boldsymbol{X}) < 0\}$. The RMAS algorithm emphasizes sampling only in the neighborhood $U(\boldsymbol{X}_i, r)$ of maximum point $\boldsymbol{X}_i \in \omega$. Specifically, the RMAS algorithm only selects one sample for each $U(\boldsymbol{X}_i, r)$. The specific sampling process is shown in Algorithm 1.

Compared to the method of collecting m points with the largest residual error, the RMAS algorithm takes into account the influence of multiple maximum values (Fig. 1). It does not require additional parameters for finding extreme points and exhibits stronger adaptability.

The guiding principle of the RMAS algorithm is to extract the maximum information with the minimum number of points, considering each maximum of the residual probability density function with minimal cost. More critically, the RMAS algorithm is a fully adaptive sampling algorithm that does not depend on any hyperparameters. This means that the RMAS algorithm can be directly applied to different tasks without spending effort on parameter tuning.

It is worth noting that when the degrees of freedom of the system are greater than 2, the probability density function of residuals will be defined in a high-dimensional space. The RMAS algorithm can utilize gradients to find the maximum value of the residual probability density function, but this process becomes significantly difficult. In certain cases, there may not be a maximum value, and it may be necessary to define points similar to saddle points. The specific scenario is discussed in Sec. IV. The system degrees of freedom for the experimental cases are all less than or equal to 2.

D. Few-shot identification framework

The stochastic physics-informed neural ordinary differential equation (SPINODE) framework successfully identifies the system's governing equations (Itô form) from a large amount of trajectory data.[14] A new few-shot identification (FSI) method for stochastic dynamical systems based on the neural network and RMAS methods is proposed in this paper. The SPINODE framework trains the neural network with all the training data at once. However, the FSI method utilizes the RMAS algorithm to selectively train the neural network with samples in multiple rounds of training, choosing the most valuable samples to expand the training set of the neural network.

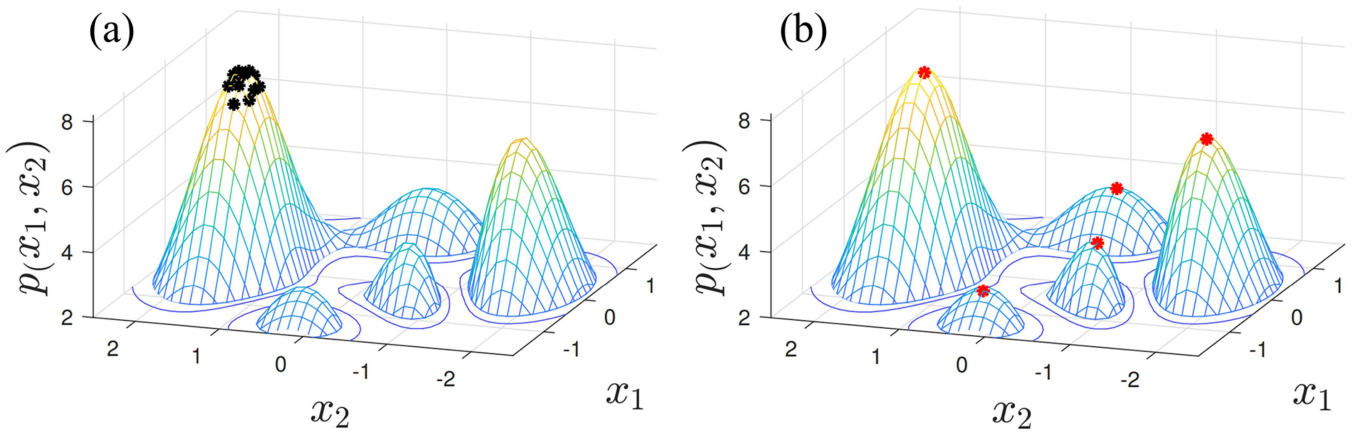

FIG. 1. Comparison of sampling strategies between the classical RAR algorithm and the RMAS algorithm. (a) RAR, $m = 9$; (b) RMAS. The mark $*$ indicates the sampling position.

The processes of the FSI method are shown in Fig. 2. The training set of the first epoch of the neural network training process comes from several sample points randomly selected in the sample space Ω. The RMAS algorithm and the neural network training are executed alternately. The RMAS algorithm generates the data required for each epoch of the neural network training process. The training results of the neural network provides the prerequisite for the RMAS algorithm. Such repeated operations will not stop until

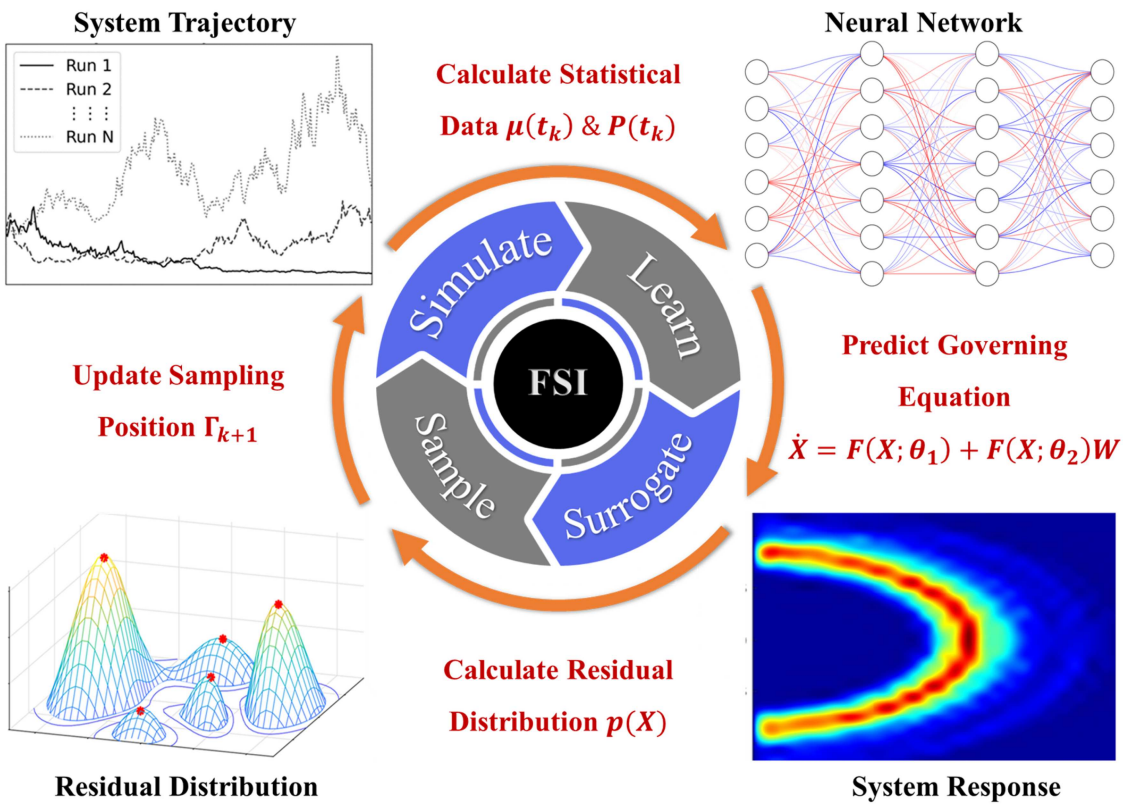

FIG. 2. Framework diagram for few-shot identifying stochastic dynamical systems.

the error (RMSE, RRMSE, etc.) of the neural network training result is less than a preset threshold.

The root mean square error (RMSE) and the relative root mean square error (RRMSE) are used to compute the training error of the neural network at different scales as follows:

$$\mathrm{RMSE} = \frac{1}{\sqrt{n\,|\Omega_0|}} \sum_{X\in\Omega_0} \left(\|F(X;\theta_1) - \Phi(X)\|_2 + \|F(X;\theta_2) - \Psi(X)\|_2 \right), \tag{14}$$

$$\mathrm{RRMSE} = \frac{1}{\sqrt{n\,|\Omega_0|}} \sum_{X\in\Omega_0} \left(\left\| \frac{F(X;\theta_1) - \Phi(X)}{\Phi(X) + \rho} \right\|_2 + \left\| \frac{F(X;\theta_2) - \Psi(X)}{\Psi(X) + \rho} \right\|_2 \right), \tag{15}$$

where ρ is a small value, for example, 10^{-6}. In addition to RMSE and RRMSE, the FSI framework can have more complex stopping criteria, for example, restrictions on the sample size.

III. RESULT

In order to verify the effectiveness of the FSI method, two examples, a vegetation biomass change model and the Rayleigh–Van der Pol impact vibration model, are selected for system identification in this paper. These two examples involve cases with one-dimensional and two-dimensional systems and have rich dynamical behaviors that can be used for an analysis (first escape probability and stochastic P bifurcation). The accurate prediction of the dynamical behavior provides a strong evidence that the method is effective.

A. Vegetation biomass change model in grazing systems

The classic vegetation biomass dynamical model in grazing systems is defined by equation $\dot{x} = sx(a-x) - q\frac{cx^2}{x^2+x_0^2}$, where s, a, q, c, and x_0 are all system parameters. This system has two steady states.[25] The transition of the system's state between the two attractors may lead the system toward different equilibrium states, which can have a significant impact on human production. Therefore, it is valuable to identify and analyze the first escape probability of the system model.

There are unavoidable random factors in the real ecological environment, so parametric noise is added to the model as equation $\dot{x} = sx(a-x) - q\frac{cx^2}{x^2+x_0^2} + x\xi(t)$. The solution of the system may jump between both domains of attraction when the noise of the parameters is taken into account.

By adding the Wong-Zakai correction term, an Itô stochastic differential equation in the form of $dx = f(x)dt + h(x)dB(t)$ is obtained. $B(t)$ is a unit Wiener process, and the drift and diffusion terms are defined as follows:

$$\begin{aligned} f(x) &= sx(a-x) - q\frac{cx^2}{x^2+x_0^2} + Dx, \\ h(x) &= \sqrt{2D}x. \end{aligned} \tag{16}$$

The system state x and the noise intensity D are treated as independent variables for the drift term and the diffusion term, which are defined in the domain $(x, D) \in [2.1, 5] \times [0.001, 0.12] = \Omega \subset R^2$. The space Ω is divided into a uniform grid of 40×40. The initial values of each node in the grid are used to solve $dx = f(x)dt + h(x)dB(t)$ by utilizing the Euler–Milstein method [Eq. (4)].

Some system parameters and simulation parameters that appear in Sec. II and Eq. (16) are set as $\Delta t = 0.01, L = 25$, $N = 10\,000$,[14] $\alpha = 10^{-3}, \beta = 2, \lambda = 1$, $s = 0.125, a = 8, q = 1.9$, $c = 1, and\ x_0 = 1$.[25] The neural network that represents $F(\theta_1)$ and $F(\theta_2)$ is comprised of two intermediate layers, each consisting of 150 dimensions. The neural network is initialized and then trained for 10 epochs in each cycle as shown in Fig. 2.

This system has two regions of attraction when the parameters are chosen as above. The region of attraction within the range of $D_1 = (0.74, 2.10)$ is named low vegetation attraction and the regions

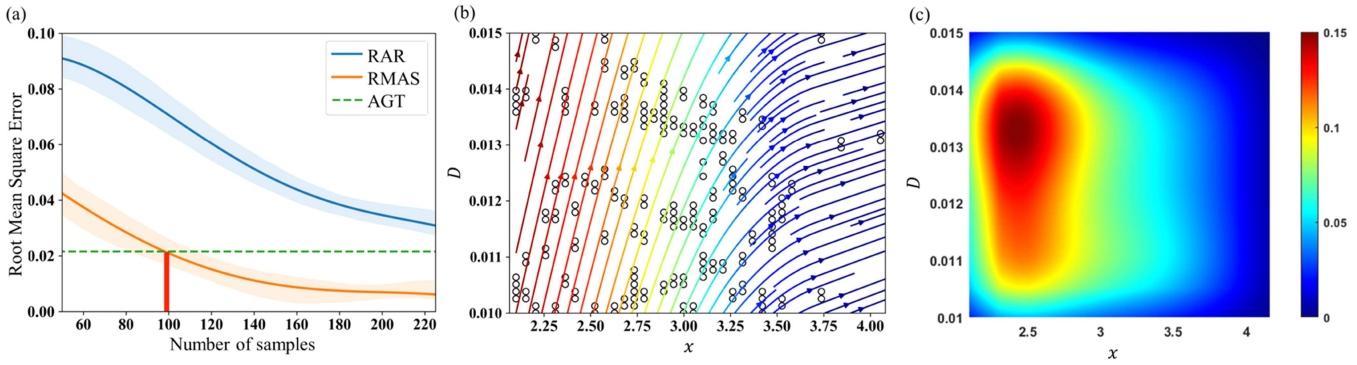

FIG. 3. Analysis of error and sampling distribution after fitting grazing system with RMAS algorithm. (a) Comparison of the training results of RMAS algorithm and RAR algorithm, where AGT is the result of training using 40×40 (uniform grid) sample points as a reference. Each algorithm is repeatedly trained 10 times. The solid line in each curve represents the average of the training results, and the boundary of the area is one standard deviation of the training results. The red line is used to highlight the number of samples required by the RMAS algorithm to achieve the desired accuracy; (b) the direction field of the drift term in Eq. (16) is labeled with all sampling positions from the RMAS algorithm. The red arrows are used to indicate a larger slope, while the blue arrows indicate the opposite; (c) cumulative residual distribution from the RMAS algorithm.

of attraction within the range of $D_2 = (2.10, 5.16)$ are named high vegetation attraction.[25]

AGT indicates training using all sample points in the grid. Compared to the RAR algorithm (which requires 250 points), the RMAS algorithm (which requires 100 points) uses much fewer sample points to make the training process achieve the desired result (AGT) [Fig. 3(a)]. In order to achieve the same level of accuracy (error threshold), the RMAS algorithm requires a sample size of approximately 10% of AGT. The RMAS algorithm tends to sample where the slope of the function being fitted is large [Fig. 3(b)]. The cumulative residuals are also large in the region where the slope of the fitted function is larger [Fig. 3(c)]. This implies that the governing equations of the system are more difficult to be identified in the regions where the system state changes rapidly.

The neural network combined with the RMAS algorithm demonstrates a good performance in system identification as shown in Fig. 4 when three different intensities D of noise are applied to the system. Although the primary method used in the fitting task is not ours, when combined with the RMAS algorithm, the same effect can be achieved in system identification with only a much smaller number of samples.

To test the accuracy and usefulness of the trained neural network models, the neural networks are used as surrogate models for $dx = f(x)dt + h(x)dB(t)$ for dynamical analysis. The first escape probability (FEP) $P_E(x)$ from the high vegetation attraction to the low vegetation attraction can be calculated by Eq. (17).[26] The numerical solutions to Eq. (17) are calculated using the difference method,

$$P_E(x) = \int C_1 e^{\int \frac{-2m(x)}{\sigma^2(x)} dx} dx + C_2. \tag{17}$$

The true FEP and the predicted FEP are very similar (Fig. 5), with an RRMSE of 0.0159. The neural network trained using the RMAS algorithm can be used as surrogate functions for dynamical analysis.

B. Rayleigh–Van der Pol impact vibration model

In recent decades, in engineering mechanics, applied mathematics, applied physics, and other fields, various collision systems have received extensive attention.[27] The Rayleigh–Van der Pol collision vibration system under parameter excitation is considered in this paper as follows:

$$\begin{aligned} \ddot{x} - \left(s - ax^2\right)\dot{x} + q\dot{x}^3 + x = x\xi_1(t) + \dot{x}\xi_2(t), \quad x > 0, \\ \dot{x}_+ = -r\dot{x}_-, \quad x = 0, \end{aligned} \tag{18}$$

where the parameters s, a, and q are real numbers and $\xi_i(t)(i = 1, 2)$ are independent Gaussian white noise. The collision occurs at the position where the displacement is zero.

By adding the Wong-Zakai correction term, Eq. (18) can be transformed into an Itô stochastic differential equations in the form,

$$\begin{aligned} dx_1(t) &= f_1(x_1, x_2)\,dt + h_1(x_1, x_2)\,dB(t), \\ dx_2(t) &= f_2(x_1, x_2)\,dt + h_2(x_1, x_2)\,dB(t), \end{aligned} \tag{19}$$

where $B(t)$ is a unit Wiener process, and the drift and the diffusion terms are defined as follows:

$$\begin{aligned} f_1(x_1, x_2) &= x_2, \\ h_1(x_1, x_2) &= \sqrt{2D_1}x_1, \\ f_2(x_1, x_2) &= (s - ax_1^2)x_2 - qx_2^3 - x_1 + Dx_2, \\ h_2(x_1, x_2) &= \sqrt{2D_2}x_2. \end{aligned} \tag{20}$$

The Gaussian noise intensities are D_1 and D_2. The system states x_1 and x_2 are treated as independent variables for the drift term and the diffusion term, which are defined in the domain $(x_1, x_2) \in [1, 3] \times [-3, -1] = \Omega \subset R^2$. The space Ω is divided by a uniform grid of 40×40. The initial values of each node in the grid are used to solve Eq. (19) by utilizing the Euler–Milstein method [Eq. (4)]. Some system parameters and simulation parameters that

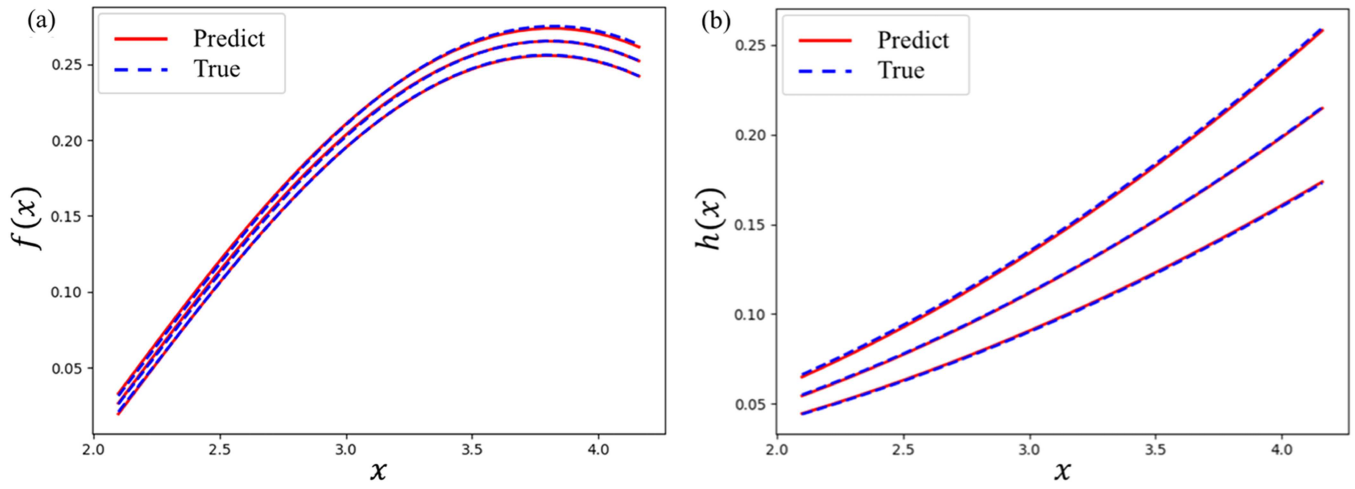

FIG. 4. Learning drift and diffusion terms for grazing system using RMAS algorithm. (a) Drift term; (b) diffusion term. The three curves from top to bottom represent the cases of $D = 0.015$, $D = 0.013$, and $D = 0.010$. The prediction within the training interval Ω cannot reflect the generalization effect of the algorithm.

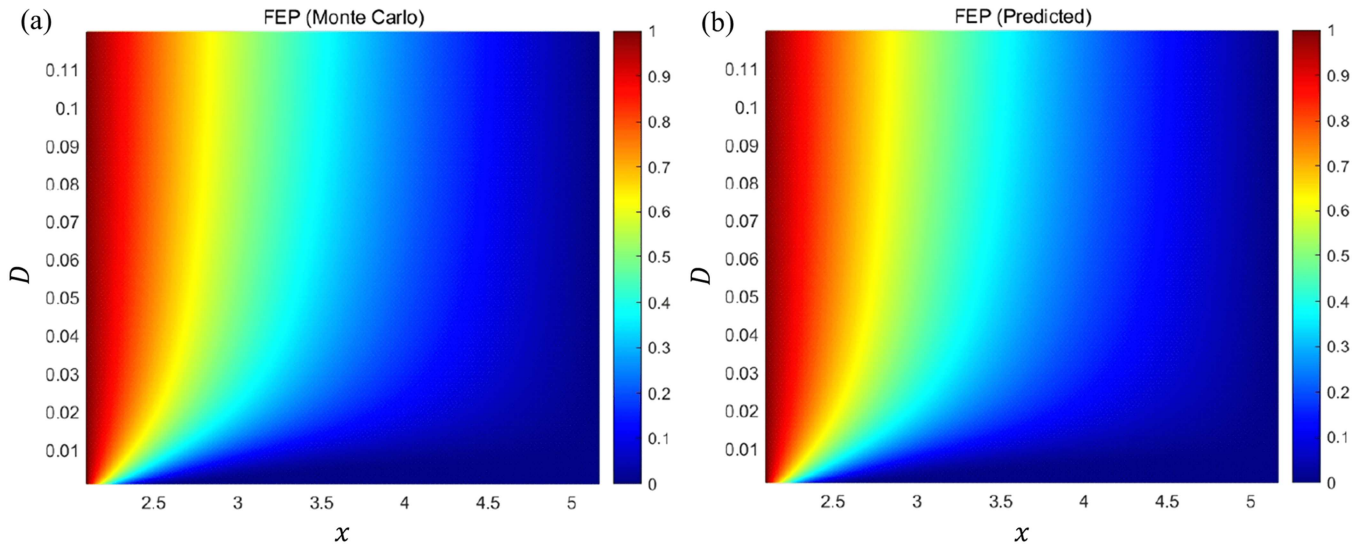

FIG. 5. Top view of first escape probability density function based on analytical method and Monte Carlo simulation. (a) The true first escape probability density function is obtained by Monte Carlo simulation of equation $\dot{x} = sx(a-x) - q\frac{cx^2}{x^2+x_0^2} + x\xi(t)$; (b) the predicted first escape probability density function is obtained by solving Eq. (17) with the neural networks $F(x;\theta_1)$ and $F(x;\theta_2)$ as surrogate functions for $m(x)$ and $\sigma(x)$.

appear in Chapter II and Eq. (18) are set as $\bigwedge t = 0.01, L = 25$, $N = 10\,000$,[14] $\alpha = 10^{-3}, \beta = 2, \lambda = 1$,[28] $s = 0.03, a = 0.01, q = 0.01$, $r = 0.98, and\ D_1 = D_2 = 0.015$.[29] The neural network that represents $F(\theta_1)$ and $F(\theta_2)$ is comprised of two intermediate layers, each consisting of 150 dimensions. The neural network is initialized and then trained for 10 epochs in each cycle (Fig. 2).

Compared to the RAR algorithm (which requires 1000 points), the RMAS algorithms (which requires 465 points) can use fewer sample points to make the training process achieve the desired result (AGT) [Fig. 6(a)]. In order to achieve the same level of accuracy (error threshold), the RMAS algorithm requires a sample size of approximately 30% of AGT. The RMAS algorithm tends to sample where the slope of the function being fitted is large [Fig. 6(b)]. The cumulative residuals are also large in the region where the slope of the fitted function is larger [Fig. 6(c)]. This implies that the governing equations of the system are more difficult to be identified in the regions where the system state changes rapidly. This is consistent with what is observed in the grazing system.

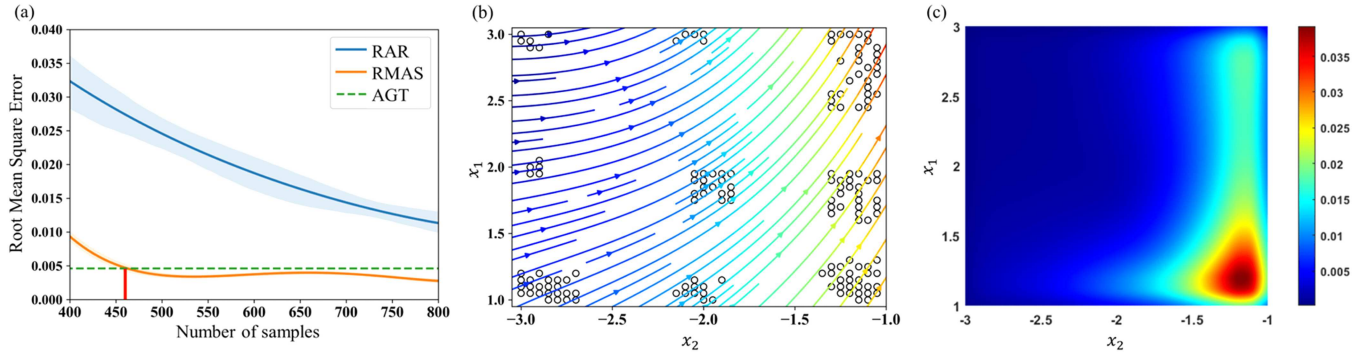

FIG. 6. Analysis of error and sampling distribution after fitting Rayleigh–Van der Pol system with RMAS algorithms. (a) Comparison of the training results of RMAS algorithms and RAR algorithm, where AGT is the result of training using 40×40 (uniform grid) sample points as a reference. Each algorithm is repeatedly trained 10 times. The solid line in each curve represents the average of the training results, and the boundary of the area is one standard deviation of the training results. The red line is used to highlight the number of samples required by the RMAS algorithm to achieve the desired accuracy; (b) the direction field of the drift term in Eq. (20) is labeled with all sampling positions from the RMAS algorithm. The red arrows are used to indicate a larger slope, while the blue arrows indicate the opposite; (c) cumulative residual distribution from the RMAS algorithm.

The neural network combined with the RMAS algorithm can accurately identify the system equations within the specified interval (Fig. 7). It is feasible to use the neural network function instead of the drift term and the diffusion term for the dynamical analysis of the system.

When the noise intensity D_2 changes from 0.005 to 0.05, the shape of the steady-state probability density function changes from a semicircle to a point, which is called P bifurcation [Figs. 8(a) and 8(c)]. The phenomenon of P bifurcation is reproduced using the neural network as a surrogate model [Figs. 8(b) and 8(d)]. These two results are very close with an RRMSE of 0.0419. This indicates that the stochastic differential equations identified with the RMAS algorithm are effective to compute the steady-state probability density function and observe the bifurcation behavior.

IV. DISCUSSION

A. Initial value perturbation

The RMAS algorithm does not need to do sensitivity analysis for hyperparameters because it has no hyperparameters. Solving inverse problems in dynamics using neural network usually requires a large amount of snapshot data of the unknown system. The trajectory data of the system usually contain noise and disturbances in practical engineering mechanics or climate fields as $\dot{x} = f(t, x, \varepsilon)$, $x(t_0) = \eta(\varepsilon)$. To eliminate and measure the effects of disturbances, various methods have been developed, including perturbation, Bayesian, and uncertainty quantization methods.[30–33] With the development of embedding physical information into neural network and the popularization of big data technology, using RMAS algorithm to mitigate the impact of noise on dynamical

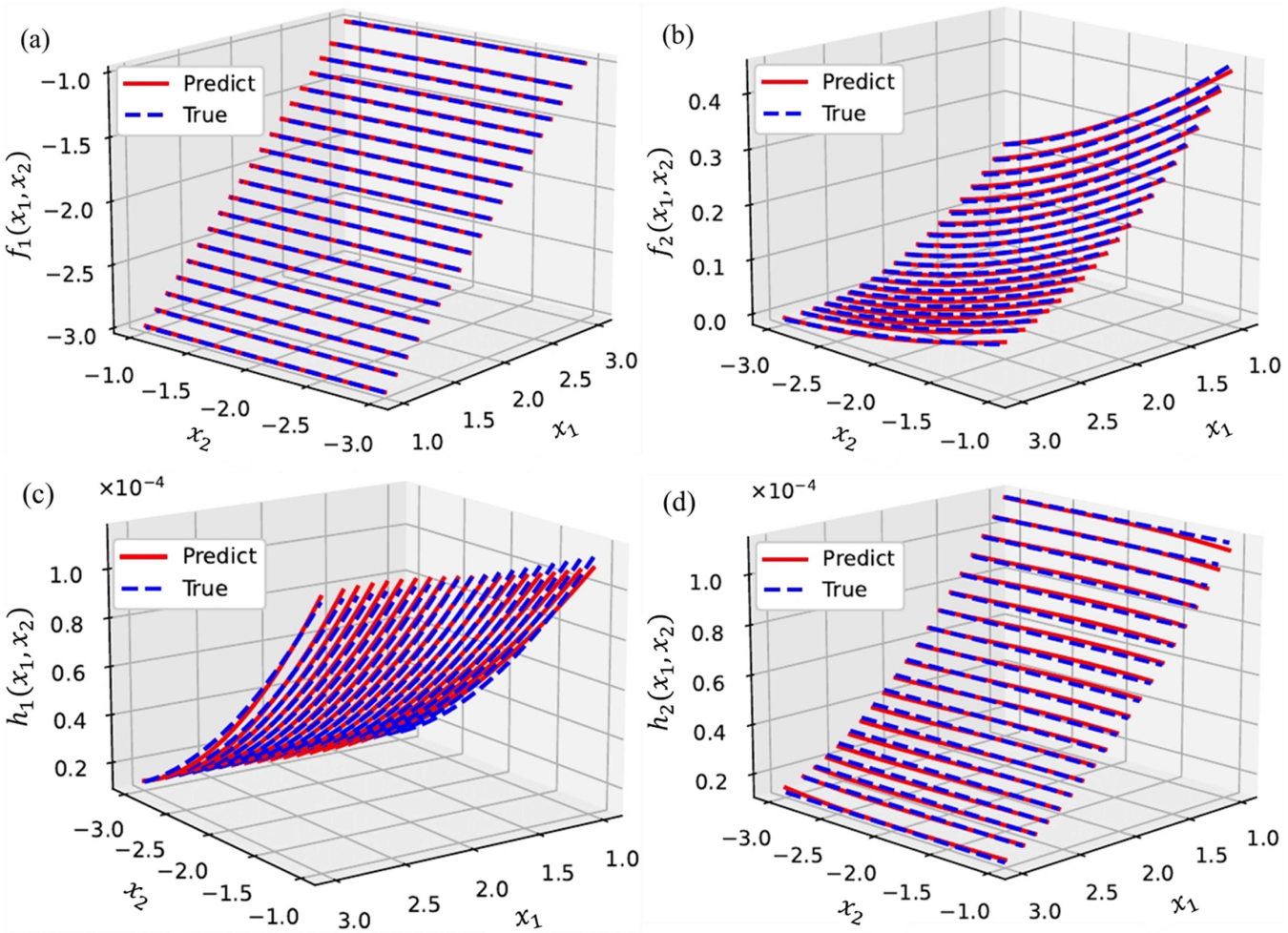

FIG. 7. Learning drift and diffusion terms for Rayleigh–Van der Pol system using RMAS algorithm. (a) and (b) are drift terms. (c) and (d) are diffusion terms. The prediction within the training interval Ω cannot reflect the generalization effect of the algorithm.

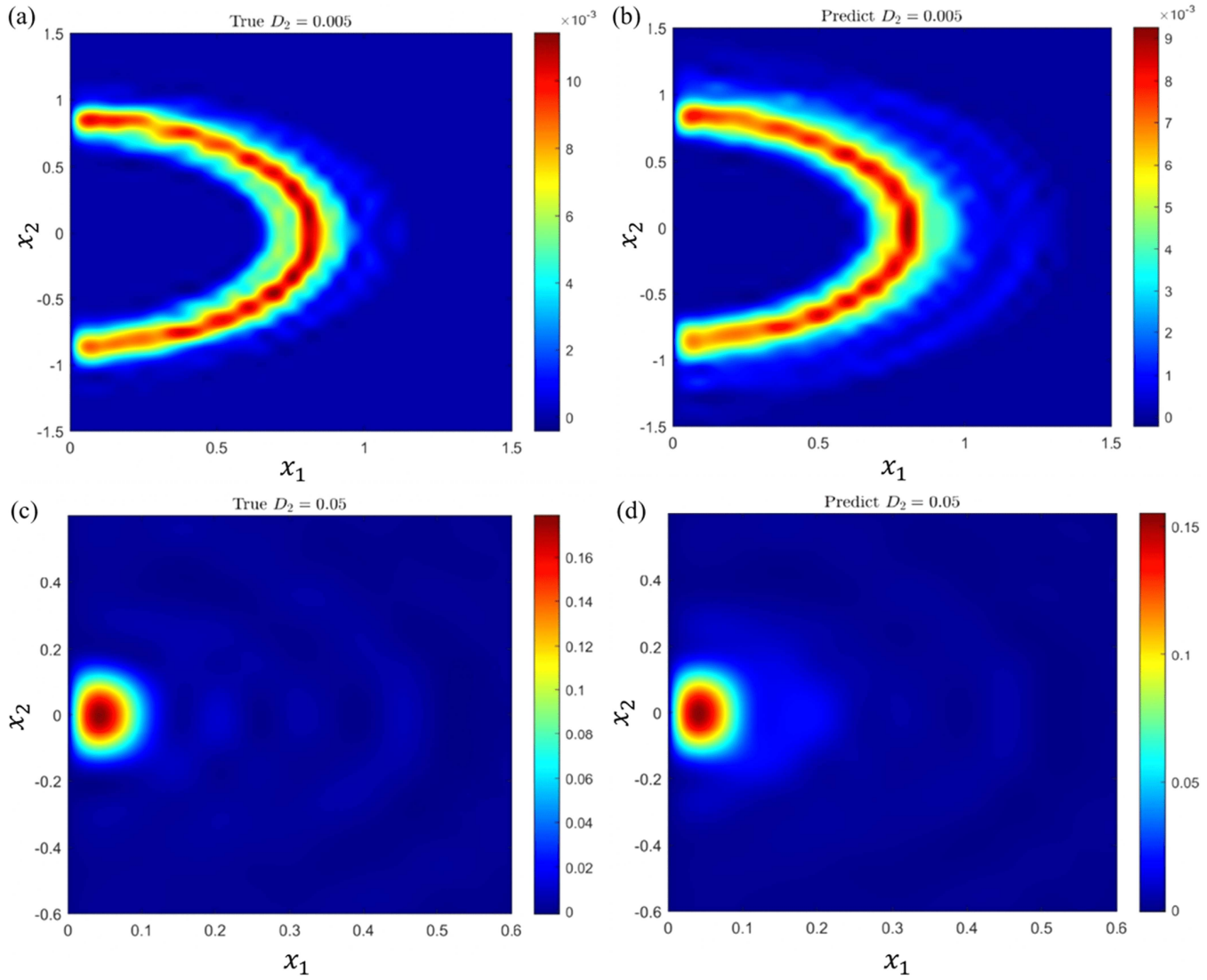

FIG. 8. Predicting P-bifurcation behavior of Rayleigh–Van der Pol system using neural network functions. The probability density functions are derived from Monte Carlo simulations of Eq. (19). The data labeled as **True** are derived from the original Eq. (19). The drift and diffusion terms in Eq. (19) are replaced by $F(\boldsymbol{x}, \theta_1)$ and $F(\boldsymbol{x}, \theta_2)$ in the data labeled as **Predict**. (a) True, $D_2 = 0.005$; (b) Predict, $D_2 = 0.005$; (c) True, $D_2 = 0.05$; (d) Predict, $D_2 = 0.05$.

system inverse problems is a novel and important perspective perceived in this paper.

Neural networks combined with the RMAS algorithm are highly dependent on data from 10 000 simulations starting from the same initial values when identifying system equations. The true initial values of two simulations are slightly different in practical engineering, as the ancient Greek philosopher Heraclitus said "one cannot step into the same river twice." The initial value perturbation problem as shown in Eq. (21) is considered in this paper,

$$\dot{x} = f(t, x, 0),\ x(t_0) = \eta(\varepsilon). \tag{21}$$

Disturbed initial values can introduce errors in both the drift and the diffusion coefficients. The statistical characteristics (moment information) extracted from multiple trajectory data starting from the same initial value may become inaccurate. The trajectory data generated by repeating simulations (10 000 times) from perturbed initial values are used to analyze the impact of initial value perturbations on moment information. In this case, the magnitude of the noise intensity within the dynamical system is much smaller than the magnitude of the state variables. Therefore, in relation to the moment information itself, the impact of initial value perturbations on the mean is small but significant on the variance [Fig. 9(a)].

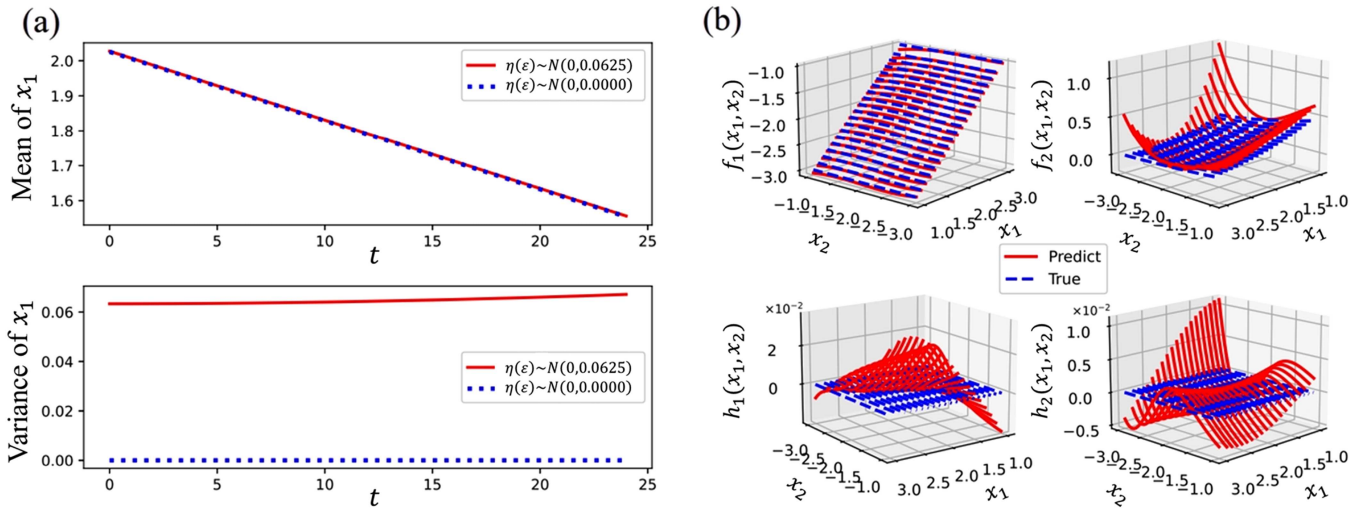

FIG. 9. The effect of initial value perturbation on the identification of Rayleigh–Van der pol system. (a) The effect of initial value perturbations on moment information; initial values were set to $x_1 = 2.1$ and $x_2 = -1.9$. The perturbations obeyed the distribution $N(0, 0.25^2)$. Equation (19) is solved 10 000 times repeatedly at 25 steps each time; (b) the effect of initial value perturbation on system identification. The system parameters and simulation parameter settings are consistent with those of the AGT in Sec. III.

The result of the system identification is bad especially the diffusion term is shown in Fig. 9(b). It is very important to mitigate the adverse effect of the initial value perturbation on the system identification.

Three metrics, including RMSE, RRMSE, and M_h, are used to measure the effectiveness of system identification. In order to measure the effect of the initial value perturbation on the diffusion term identification, we introduce a new metric M_h defined as

$$M_h = \max\{\|F(X;\theta_2) - \Psi(X)\|_1 \,|X \in \Omega_0\}. \tag{22}$$

We show that the RMAS algorithm requires only 300 sample points to achieve the training result of AGT (1600) in system identification for the Grazing system [Fig. 10(a)]. The required number of samples is 195 for the Rayleigh–Van der Pol system as shown in Fig. 10(b). The RMAS algorithm mitigates the bad effect of trajectory data containing initial value perturbations on system identification by selecting a small number of samples.

The Rayleigh–Van der Pol system exhibits higher sensitivity to initial value perturbations than the Graz system. This is supported by the much larger value of RRMSE in Fig. 10(b) compared to Fig. 10(a). The RMAS algorithm can also achieve training results compare to AGT with much smaller samples when the identified system is more sensitive to perturbed trajectory data.

When the noise intensity is small or large, the RMAS algorithm requires only a small number of sample points to achieve the same accuracy as AGT in all three metrics as shown in Fig. 11. The effect

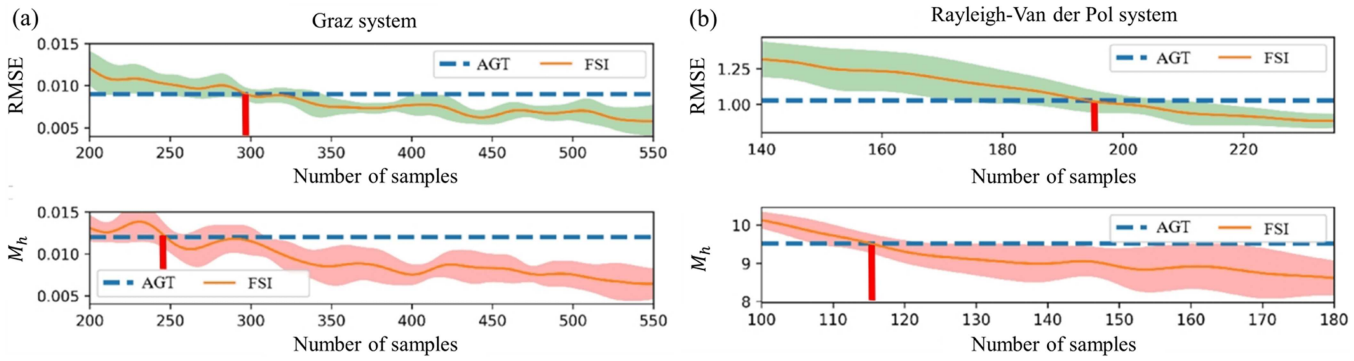

FIG. 10. The variations in the three indicators in the system identification tasks of the Graz system and Rayleigh–Van der Pol system. (a) Graz system. (b) Rayleigh–Van der Pol system. The system parameters and simulation parameters used in the experiment are consistent with those in Sec. III. RMAS algorithm is repeatedly trained 10 times. The solid line represents the average of the training results, and the boundary of the area is one standard deviation of the training results.

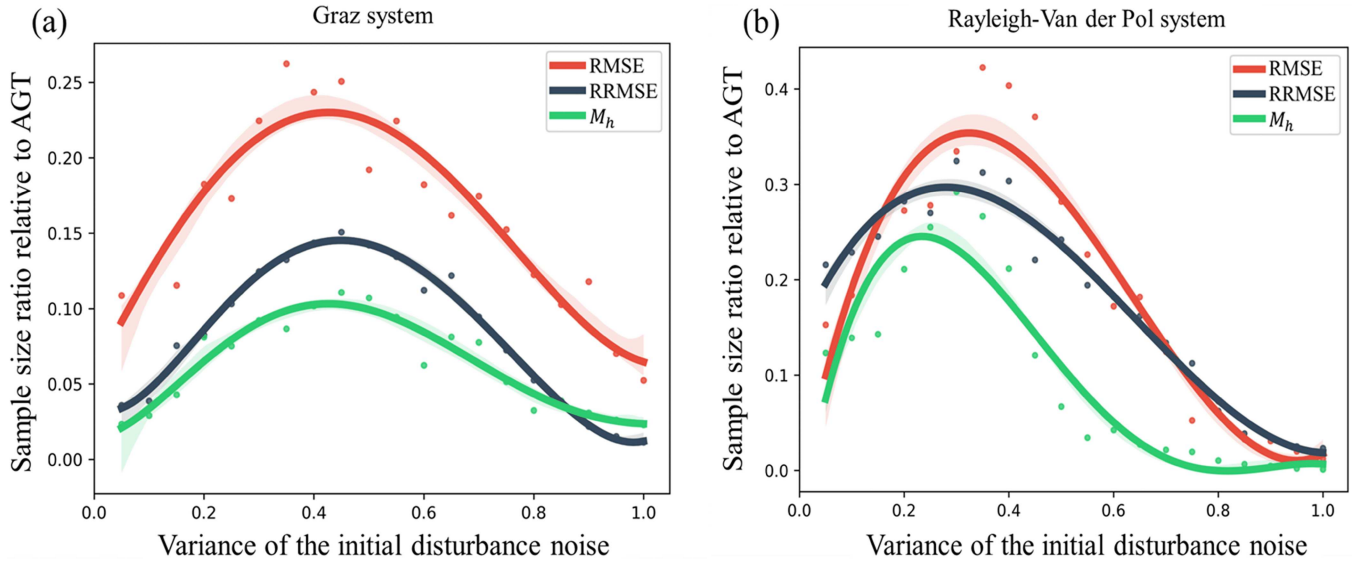

FIG. 11. The number of samples required for RMAS algorithm to reach the AGT training results for different strengths of the initial value perturbation under the three metrics. (a) Graz system. (b) Rayleigh–Van der Pol system. The system parameters and simulation parameters used in the experiment are consistent with those in Sec. II. The solid scatter indicates the mean value of the results of 10 replicate experiments. The fitted curves of the scatter points are bounded by a 95% confidence interval.

of the RMAS algorithm is not significant when the trajectory data of the system contains moderately initial value perturbation noise [about $N(0, 0.5)$].

These results indicate that the RMAS algorithm is robust in the task of system identification using deep learning methods. By using fewer samples, the RMAS algorithm can mitigate the impact of initial value perturbations on the system identification task.

B. Lack of residual information

The RMAS algorithm utilizes residual information to select efficient sampling locations, thereby saving at least 70% of the data collection cost (for directly simulating the Itô equation). Undoubtedly, the more accurate the residual probability density function, the better the performance of the RMAS algorithm. However, obtaining an accurate residual probability density function is challenging in practical engineering applications. Therefore, it is necessary for this paper to discuss the performance of the RMAS algorithm when there is insufficient prior information. Let M_r represent the proportion of missing points with insufficient prior information on the drift term $\Phi(\boldsymbol{X})$ and diffusion term $\Psi(\boldsymbol{X})$ within the domain I^n, relative to the total number of points in a sampling grid space.

In order to perform system identification experiments on the vegetation biomass change model and Rayleigh–Van der Pol impact vibration model, this study conducted the experiments five times to ensure data adequacy and result stability while keeping other variables constant.

Figure 12 demonstrates that for small sample sizes (less than 50), the identification results deteriorate as the number of points with missing prior information increases. This is primarily due to the lack of prior information, which results in inaccurate computation of the residual function. As a result, the RMAS method fails to find the most effective sampling points, leading to larger errors in the surrogate function. It is worth noting that as the sample size gradually increases, the drawbacks of missing prior information become less pronounced. This is because even though the RMAS method does not find the most effective sample points in each iteration, when the sample points accumulate sufficiently, the number of sample points reaches a "saturation" state, where the surrogate function errors for different M_r values are at the same level.

With repeated execution of the RMAS algorithm, the sample size gradually increases, and the influence of the error in the residual probability density function on the results diminishes, resulting in the convergence of system identification accuracy to a stable level. Additionally, the required sample size for the RMAS algorithm (82 and 280) remains significantly lower than the total sample size (1600). Therefore, this study concludes that RMAS exhibits good robustness in the presence of missing prior information. The negative impact of missing prior information on the results can be mitigated by slightly increasing the sample size (performing additional rounds of the RMAS algorithm). However, it is undeniable that RMAS is still an efficient sampling algorithm.

C. High-dimensional dynamical system

Few-shot identification can indeed be applied to high-dimensional systems, including ODE systems arising from the discretization of stochastic partial differential equations. For example, in the field of nonlinear manifold learning, machine learning techniques such as clustering and autoencoder neural network can be used to learn the underlying low-dimensional dynamics.[34–36] The residual-based multipeaks adaptive sampling (RMAS) method can

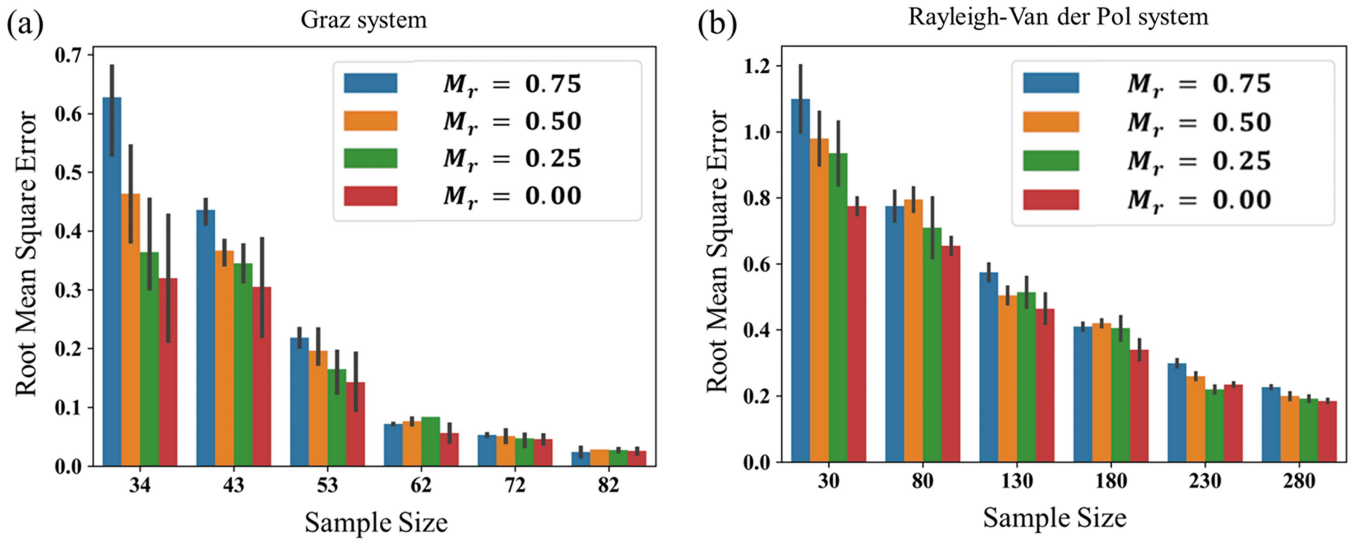

FIG. 12. Graph of the variation in sample size and error under the condition of missing prior information. M_r represents the proportion of points with missing prior information on the drift term and diffusion term within the domain I^n, relative to the total number of points. (a) Grazing System. (b) Rayleigh–Van der Pol System.

be directly applied when manifold learning involves time snapshot data of the system and neural network. The entire process is illustrated in Fig. 2. The RMAS algorithm can provide sampling suggestions when any residual distribution can be obtained.

One specific concern is how to adaptively search for extremal points of high-dimensional functions. The first-order and second-order derivatives can be used to find the maximum points of low-dimensional ($n \leq 2$) functions. However, when the dimensionality of the system is greater than or equal to 3, the increasing dimensionality of the residual probability density function $p(X)$ makes it challenge to compute the maximum points in the RMAS method. To overcome this challenge, this paper presents the definition of m-order saddle points.

Definition 1. For a domain $I^n = \left[x_L^1, x_U^1\right] \times \left[x_L^2, x_U^2\right] \times \cdots \times \left[x_L^n, x_U^n\right]$, an n-dimensional function $f\left(x^1, x^2, \ldots, x^n\right)$ is defined. For a point $\left(x_0^1, x_0^2, \ldots, x_0^n\right) \in I^n$, if there exist m elements $x_0^i \in \left\{x_0^i\right\}$, and each x_0^i satisfies that $\left(x_0^1, x_0^2, \ldots x_0^i, \ldots, x_0^n\right)$ is a local maximum point of the one-dimensional function $f\left(x_0^1, x_0^2, \ldots, x^i, \ldots, x_0^n\right)$, where $x^i \in \left[x_L^i, x_U^i\right]$, then the point $\left(x_0^1, x_0^2, \ldots, x_0^n\right)$ is called a m-order saddle point of the n-dimensional function $f\left(x^1, x^2, \ldots, x^n\right)$. If $m = n$, then the point $\left(x_0^1, x_0^2, \ldots, x_0^n\right)$ is called a local maximum point of the n-dimensional function $f\left(x^1, x^2, \ldots, x^n\right)$.

The definition of an m-order saddle point considers the values of extreme points in each dimension of an n-element function $f(x^1, x^2, \ldots, x^n)$. A point that is an extreme point in more dimensions is more likely to be an extreme point of the multivariate function over its entire domain. Typically, an n-element function $f(x^1, x^2, \ldots, x^n)$ does not necessarily have a global maximum, but it may have several saddle points of different orders.

The definition of an m-order saddle point involves finding the maximum points of a high-dimensional residual function $p(X)$. The procedure is as follows: for an n-dimensional PDF of the residual $p(X)$, first calculate its n-order saddle points. If no solution is found, calculate the $(n-1)$-order saddle points, and so on, until the set of m-order saddle points with $0 < m < n$ is non-empty. Then, select all m-order saddle points from this set as new sampling positions. Additionally, this is a fully adaptive algorithm.

The definition of a m-order saddle point ensures that in each iteration of the FSI method, the RMAS method can always obtain new sampling positions. This is because, even in the most extreme case, any non-constant $p(X)$ will have several first-order saddle points (including interval endpoints). One important detail to note is that in the main loop of FSI, if a selected new sampling point $X_0 = (x_0^1, x_0^2, \ldots, x_0^n)$ already exists in the existing set of sampling points, the new sampling point should be chosen from the set of points closest to X_0 in terms of Euclidean distance, with the condition that it has the largest residual and is not already part of the existing set of sampling points.

V. CONCLUSION

In this paper, in order to reduce the demand of data for system identification, a residual-based multi-peak adaptive sampling (RMAS) method is proposed, which is a fully adaptive sampling algorithm without hyperparameters. Subsequently, this paper combines the RMAS method and neural network to propose a small sample identification (FSI) method for stochastic dynamical systems. The FSI method gradually inputs the sample points into the neural network for training through iteration.

We have used the FSI method for two stochastic differential equations for the system identification task in the case study section. After the training process, the trained neural network can be used as a surrogate function to analyze the stochastic dynamical behavior of the system. Our results show that the neural network combined with the RMAS algorithm requires only 20%–30% of the original sample

size (uniform grid sampling) to get an accurate surrogate model. The RMSE of the surrogate model prediction is less than 2×10^{-2}. The surrogate model accurately predicts the first escape probability density function and the P bifurcation behavior in the systems with an RMSE of less than 1.6×10^{-2}.

The robustness of the FSI method is tested by adding an initial value perturbation to the trajectory data. The initial value perturbation that follows a normal distribution with mean zero has a detrimental effect on the identification of the diffusion term. A new metric, named M_h, is defined to measure this effect. The FSI method can effectively alleviate the overfitting of neural network to initial value perturbation as demonstrated by the RMSE, RRMSE, and M_h. The effect of the FSI method is not significant when the system trajectory data contain moderately strong initial perturbation noise [approximately $N(0, 0.5)$]. Therefore, the FSI method is a robust few-shot learning algorithm.

The current sampling algorithm only determines residuals and sampling positions based on the training results of one epoch of the neural network. In future research, the training results of multiple epochs of the neural network will be used to develop more advanced sampling strategies. Additionally, improving the generalization ability of neural network as surrogate model through reasonable sampling is future research direction.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (NNSFC) (Nos. 12172291 and 12232015), Shaanxi Province Outstanding Youth Fund Project (No. 2024JC-JCQN-05), and the 111 Project (No. BP0719007).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Devised the research project: Xiao-Kai An, Lin Du, and Zi-Chen Deng. Theoretical derivation and performed numerical experiments: Xiao-Kai An and Lin Du. Analyzed the results: Xiao-Kai An and Yu-Jia Zhang. Wrote the paper: Feng Jiang and Lin Du. All authors reviewed the manuscript.

Xiao-Kai An: Conceptualization (lead); Data curation (lead); Formal analysis (lead); Investigation (lead); Methodology (lead); Software (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (lead). **Lin Du:** Project administration (equal); Resources (equal); Supervision (equal); Validation (equal); Writing – review & editing (lead). **Feng Jiang:** Investigation (equal); Resources (equal); Validation (equal); Writing – review & editing (supporting). **Yu-Jia Zhang:** Investigation (supporting); Resources (supporting); Writing – review & editing (supporting). **Zi-Chen Deng:** Funding acquisition (lead); Supervision (lead); Writing – review & editing (equal). **Jürgen Kurths:** Supervision (lead); 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]Q. Liu, Y. Xu, J. Kurths, and X. Liu, "Complex nonlinear dynamics and vibration suppression of conceptual airfoil models: A state-of-the-art overview," Chaos **32**, 062101 (2022).
[2]J. Kou and W. Zhang, "Data-driven modeling for unsteady aerodynamics and aeroelasticity," Prog. Aerosp. Sci. **125**, 100725 (2021).
[3]S. Brunton, B. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," Annu. Rev. Fluid Mech. **52**, 477–508 (2020).
[4]S. L. Brunton, J. Nathan Kutz, K. Manohar, *et al.*, "Data-driven aerospace engineering: Reframing the industry with machine learning," AIAA J. **59**, 2820–2847 (2021).
[5]J.-P. Noël and G. Kerschen, "Nonlinear system identification in structural dynamics: 10 more years of progress," Mech. Syst. Signal Process. **83**, 2–35 (2017).
[6]J.-S. Pei and S. Masri, "Demonstration and validation of constructive initialization method for neural networks to approximate nonlinear functions in engineering mechanics applications," Nonlinear Dyn. **79**, 2099–2119 (2015).
[7]H. V. H. Ayala and L. dos Santos Coelho, "Cascaded evolutionary algorithm for nonlinear system identification based on correlation functions and radial basis functions neural networks," Mech. Syst. Signal Process. **68**, 378–393 (2016).
[8]A. R. Tavakolpour-Saleh, S. A. Nasib, A. Sepasyan, and S. M. Hashemi, "Parametric and nonparametric system identification of an experimental turbojet engine," Aerosp. Sci. Technol. **43**, 21–29 (2015).
[9]P. Dreesen, M. Ishteva, and J. Schoukens, "Decoupling multivariate polynomials using first-order information and tensor decompositions," SIAM J. Matrix Anal. Appl. **36**, 864–879 (2015).
[10]S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," Adv. Neural Inf. Process. Syst. **32**, 15379–15389 (2019).
[11]M. Finzi, K. A. Wang, and A. G. Wilson, "Simplifying Hamiltonian and Lagrangian neural networks via explicit constraints," Adv. Neural Inf. Process. Syst. **33**, 13880–13889 (2020), https://proceedings.neurips.cc/paper_files/paper/2020 and https://proceedings.neurips.cc/paper_files/paper/2020/hash/9f655cc8884fda7ad6d8a6fb15cc001e-Abstract.html.
[12]M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Comput. Phys. **378**, 686–707 (2019).
[13]R. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," Adv. Neural Inf. Process. Syst. **31**, 6572–6583 (2018).
[14]J. O'Leary, J. Paulson, and A. Mesbah, "Stochastic physics-informed neural ordinary differential equations," J. Comput. Phys. **468**, 111466 (2022).
[15]L. Lu, X. Meng, Z. Mao, and G. Karniadakis, "Deepxde: A deep learning library for solving differential equations," SIAM Rev. **63**, 208–228 (2021).
[16]C. Wu, M. Zhu, Q. Tan, Y. Kartha, and L. Lu, "A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks," Comput. Methods Appl. Mech. Eng. **403**, 115671 (2023).
[17]M. A. Nabian, R. J. Gladstone, and H. Meidani, "Efficient training of physics-informed neural networks via importance sampling," Comput.-Aided Civ. Infrastruct. Eng. **36**, 962–977 (2021).
[18]W. Gao and C. Wang, "Active learning based sampling for high-dimensional nonlinear partial differential equations," J. Comput. Phys. **475**, 111848 (2023).
[19]K. Tang, X. Wan, and C. Yang, "Das-pinns: A deep adaptive sampling method for solving high-dimensional partial differential equations," J. Comput. Phys. **476**, 111868 (2023).
[20]J. Hanna, J. Aguado, S. Comas-Cardona, R. Askri, and D. Borzacchiello, "Residual-based adaptivity for two-phase flow simulation in porous media using physics-informed neural networks," Comput. Methods Appl. Mech. Eng. **396**, 115100 (2022).
[21]S. Zeng, Z. Zhang, and Q. Zou, "Adaptive deep neural networks methods for high-dimensional partial differential equations," J. Comput. Phys. **463**, 111232 (2022).

[22] J. Paulson, M. Martin-Casas, and A. Mesbah, "Input design for online fault diagnosis of nonlinear systems with stochastic uncertainty," Ind. Eng. Chem. Res. **56**, 9593–9605 (2017).
[23] K. Ponomareva, P. Date, and Z. Wang, "A new unscented Kalman filter with higher order moment-matching," Proc. 19th Int. Symp. Math. Theory Netw. Syst. **5**, 1609–1613 (2010).
[24] D. Ebeigbe, T. Berry, M. Norton, A. Whalen, D. Simon, T. Sauer, and S. Schiff, "A generalized unscented transformation for probability distributions," arXiv (2021).
[25] H. Zhang, W. Xu, Y. Lei, and Y. Qiao, "Noise-induced vegetation transitions in the grazing ecosystem," Appl. Math. Model. **76**, 225–237 (2019).
[26] J. Duan, *An Introduction to Stochastic Dynamics* (Cambridge University Press, 2015), Vol. 51.
[27] R. Ibrahim, *Vibro-Impact Dynamics: Modeling, Mapping and Applications* (Springer Science & Business Media, 2009), Vol. 43.
[28] S. Julier and J. Uhlmann, "New extension of the kalman filter to nonlinear systems," in *Signal Processing, Sensor Fusion, and Target Recognition VI* (SPIE, 1997), Vol. 3068, pp. 182–193.
[29] P. Han, L. Wang, W. Xu, H. Zhang, and Z. Ren, "The stochastic p-bifurcation analysis of the impact system via the most probable response," Chaos Soliton Fract **144**, 110631 (2021).
[30] S. Salahshour, A. Ahmadian, B. Pansera, and M. Ferrara, "Uncertain inverse problem for fractional dynamical systems using perturbed collage theorem," Commun. Nonlinear Sci. Numer. Simul. **94**, 105553 (2021).
[31] J.-H. He, "Homotopy perturbation method: A new nonlinear analytical technique," Appl. Math. Comput. **135**, 73–79 (2003).
[32] J. Beck and L. Katafygiotis, "Updating models and their uncertainties. I: Bayesian statistical framework," J. Eng. Mech. **124**, 455–461 (1998).
[33] L. Katafygiotis and J. Beck, "Updating models and their uncertainties. II: Model identifiability," J. Eng. Mech. **124**, 463–467 (1998).
[34] D. Floryan and M. D. Graham, "Data-driven discovery of intrinsic dynamics," Nat. Mach. Intell. **4**, 1113–1120 (2022).
[35] M. Meilă and H. Zhang, "Manifold learning: What, how, and why," Annu. Rev. Stat. Appl. **11**, 393–417 (2023).
[36] A. J. Linot and M. D. Graham, "Deep learning to discover and predict dynamics on an inertial manifold," Phys. Rev. E **101**, 062209 (2020).