



# An improved sparse identification of nonlinear dynamics with Akaike information criterion and group sparsity

Xin Dong · Yu-Long Bai · Yani Lu · Manhong Fan

Received: 31 March 2022 / Accepted: 3 September 2022 / Published online: 11 October 2022  
© The Author(s), under exclusive licence to Springer Nature B.V. 2022

**Abstract** A crucial challenge encountered in diverse areas of engineering applications involves speculating the governing equations based upon partial observations. On this basis, a variant of the sparse identification of nonlinear dynamics (SINDy) algorithm is developed. First, the Akaike information criterion (AIC) is integrated to enforce model selection by hierarchically ranking the most informative model from several manageable candidate models. This integration avoids restricting the number of candidate models, which is a disadvantage of the traditional methods for model selection. The subsequent procedure expands the structure of dynamics from ordinary differential equations (ODEs) to partial differential equations (PDEs), while group sparsity is employed to identify the nonconstant coefficients of partial differential equations. Of practical consideration within an integrated frame is data processing, which tends to treat noise separate from signals and tends to parametrize the noise probability distribution. In particular, the coefficients of a species of canonical ODEs and PDEs, such as the *Van der Pol*, *Rössler*, *Burgers'* and *Kuramoto–Sivashinsky equations*, can be identified correctly with the introduction of noise. Furthermore, except for normal noise, the proposed approach is able to capture the distribution of uniform

noise. In accordance with the results of the experiments, the computational speed is markedly advanced and possesses robustness.

**Keywords** Nonlinear dynamical systems · Sparse identification · Akaike information criterion · Group sparsity · Identification of noise distribution

## 1 Introduction

Across a range of engineering fields, if the form of the governing equations is clearly known, it is possible to analyze, forecast and control dynamics. However, because it is impossible to obtain all observed data due to the complexity of the nonlinear dynamics, solving the governing equations of nonlinear dynamics has become a challenging research task. Although models are initiated from first principles [1], as the latest motivation toward machine learning, the highlighted developments posit on data-driven model discovery [2], with a much broader class of methods, including Koopman mode decomposition [3–5], dynamic mode decomposition (DMD) [6, 9], neural networks [7, 8, 12, 31], equation-free modeling [10], and nonlinear Laplacian spectral analysis [11]. Advancements regarding parsimonious models, which strike a balance between accuracy and complexity, are particularly notable.

---

X. Dong · Y.-L. Bai (✉) · Y. Lu · M. Fan  
College of Physics and Electrical Engineering, Northwest Normal University, Lanzhou 730070, Gansu, China  
e-mail: baiyulong@nwnu.edu.cn

In this line of evolution, innovative achievements utilizing symbolic regression [12] were able to directly realize nonlinear dynamics from data. Recently, sparsity-promoting technologies via regularization have been employed to robustly find the sparse representation in the space of the potential functions for nonlinear dynamics, incorporating the least absolute shrinkage and selection operator (LASSO) [13], sequentially thresholded least squares (STLSQ) [14, 24], stagewise regression [15], basis pursuit denoising (BPDN) [16], and matching pursuits [17].

Superior model discovery demands a severe technology for validation. Model selection [18–23] built on Occam’s Razor filters parsimonious, explanatory and generalizable models from an unknown feature library via Pareto analysis. Furthermore, a well-selected model can be evaluated by information loss between observed data and model-generated data via information criteria, such as the Akaike information criterion (AIC) [19] or Bayesian information criterion (BIC) [20].

The sparse identification of nonlinear dynamics (SINDy) [24] algorithm is adopted in the context of data-driven model discovery and sparse regression. Its developments are discussed in Sect. 1.1, and the contributions of this paper are outlined in Sect. 1.2.

### 1.1 Overview of SINDy

As the SINDy algorithm directly obtains the governing equations of the nonlinear dynamics through partially known data, its extensions have existed across various engineering and science fields due to its particular architecture. It frames the problem for solving governing equations of nonlinear dynamics as the sparse regression framework via a nonlinear, predefined function library. Table 1 outlines a review of the SINDy algorithm.

As seen in Table 1, the extensions of the SINDy algorithm have been classified by modified architecture and distinct applications. Control terms or external forcing are introduced to augment the potential nonlinear function library [25], even for rational nonlinearities [26], integrating the alternating direction method [27] to solve the implicit function in the null space. In addition, since *Lorenz-like systems* [28] are exceedingly sensitive to initial conditions, the dismissed terms or discontinuous points [29] may be considered.

More recently, Poincaré mappings [30], intrinsic coordinates [31], constrained physical laws [32], and

integral forms of the SINDy model [33] have been incorporated into the candidate function library to obtain well-selected models.

The process of the sparse regression problem can be recast as a convex problem. Therefore, convex relaxation regularization [34, 35] is embedded in the native SINDy frame for parameter estimation except for its inadequate robustness analysis. Bootstrap aggregating is utilized to robustify the SINDy algorithm [36]. Furthermore, spike-and-slab prior, regularized horse-shoe prior [37] or Bayesian inference [38] is considered to promote robustness for limited data. Additionally, clustering is integrated into the SINDy model to identify the turning point for infectious disease [39, 40]. The underlying idea behind the SINDy may be deeply exploited to exceedingly extended areas, such as biology [26], optics [41] and physical fluids [42, 43]. Additionally, Zhang et al. [14] provided the proof of the convergence of SINDy, and optimization techniques have been utilized to reduce the derivative error in the original SINDy algorithm [44]. Afterward, the Python package for SINDy was explored [45, 46].

### 1.2 Contributions of this work

The proposed technique in this work focuses on the improved framework of the SINDy algorithm to identify the coefficients of a class of typical nonlinear dynamics and the noise distribution. The principle dedications and innovations of this paper are illustrated as follows.

- (1) A variant method of the SINDy is developed with the Akaike information criterion to directly identify governing equations from data.
- (2) Model selection is integrated with the original SINDy framework to identify the number of terms of nonlinearities in dynamics.
- (3) The group sparsity is embedded in the sparse regression to learn the coefficients of PDEs due to the increased complexity of nonlinear dynamics.
- (4) Considering the situation in which noise exists [53], input data may be polluted by noise or other perturbation elements. To avoid generating error for derivative estimation, it is required to divide the observations into noisy estimations and estimated measurement data to

**Table 1** Overview of SINDy

Classification	Literature	Method	Modification/Field	Objective
Candidate function library	Brunton et al. [25]	SINDyc (SINDy with control)	Include external actuation and control signals	Generalize the SINDy algorithm to include inputs and control
	Mangan et al. [26]	Implicit-SINDy	Rational function nonlinearities with cross terms	Infer networked nonlinear dynamical systems
	Quade et al. [29]	Abrupt-SINDy	Either additional, deleted, or modified model terms	Learn parsimonious models of a system in response to abrupt changes
	Kaiser et al. [47]	SINDy-MPC (model prediction control)	Include the effects of actuation	Enhance data-driven control of physical systems
	Champion et al. [31]	Autoencoder-SINDy	Include intrinsic coordinates	Discover a coordinate transformation into a reduced space
	Bramburger et al. [30]	UPO-SINDy (unstable periodic orbits)	Contain parameters for Poincaré mappings	Produce parameter dependent Poincaré mappings to stabilize UPOs
	Kaptanoglu et al. [32]	Constrained-SINDy	Incorporate known physical laws	Build a connection to the large-scale Galerkin in fluid mechanics
	Wei [33]	Integral-SINDy	Contain the integral form of state equations	Identify model structure and parameters of nonlinear ODEs
Sparse regression	Hoffmann et al. [48]	Reactive-SINDy	Incorporate concentration time series	Estimate a parsimonious reaction network
	Champion et al. [35]	SR3(sparse relaxed regularized regression)	Additional regularizer for auxiliary variable	Solve a relaxation of the regularized problem
	Chu et al. [49]	Lagrangian-SINDy	Transform the naive formulation as the Legendre form	Learn interpretable and sparse formulations of overall energy and the Lagrangian
	Dai et al. [50]	Extended-SINDy	Stepwise Sparse Regressor (SSR)	Detect the dynamic behavior from data
	Fasel et al. [36]	Ensemble-SINDy	Incorporate the bootstrap aggregating (bagging)	Robustify the SINDy algorithm
	Hirsh et al. [37]	UQ-SINDy (uncertainty quantification)	Add the <i>spike-and-slab prior</i> and the <i>regularized horseshoe prior</i>	Promote robustness against noise measurements for out-of-sample forecast
Modified structure	Ram et al. [38]	Bayesian-SINDy	Combine with the neuronized priors	Handle parameter and structural model uncertainties
	Brunton et al. [51]	HAVOK (Hankel alternative view of Koopman analysis)	Integrate with delay embedding and Koopman theory	Decomposition of chaotic dynamics into linear models using forcing actuation
	Loiseau et al. [52]	SINDy-KNN (K-nearest neighbors)	Incorporate the KNN to map into the low-dimensional space	Yield interpretable nonlinear models from measurement data
	Kaheman et al. [53]	Modified-SINDy	Integrate with the automatic differential for neural networks	Denoise the data and parametrize the noise probability distribution
	Dam et al. [54]	Bifurcation-SINDy	Combine with the bifurcation analysis	Learn a underlying model from simulations of a convection system

**Table 1** continued

Classification	Literature	Method	Modification/Field	Objective
Mathematical theory	Zhang et al. [14]	SINDy	Convergence performance	Make a proof that SINDy approximates to local minimizers of an unconstrained $l_0$ -penalty least squares issue
	de Silva et al. [45]	PySINDy (Python for the SINDy)	Code package for the SINDy algorithm	A Python package for SINDy
	Messenger et al. [55]	Weak-SINDy	Formulate the problem as the weak form	Data-driven reconstruction of model coefficients from measurements
	Wu et al. [44]	SINDy	Error processing from an optimize perspective	Improve the accuracy of data
	Cortiella et al. [56]	SINDy	Consider regularized weight	Determine governing equations of nonlinear dynamics from noisy observations
Application	Jiang et al. [40]	SINDy-LM (Levenberg–Marquardt)	Disease (especially for COVID-19)	Analyze the infectious disease in both Chinese mainland and other countries
	Bhadriraju et al. [57]	SINDy	Chemistry	Achieve the adaptive model identification
	Fukami et al. [42]	CNN-LSTM-SINDy	Fluid flows	Investigate influence on the parameter choice
	Loiseau et al. [43]	SINDy	Thermal convection dynamics	Predict a bifurcation of the high-dimensional system
	Jadhav et al. [58]	SINDy	Video process	Extract related intelligence from actual videos of strongly stochastic systems

simultaneously denoise the measurements and to determine the probability distribution of the noise.

### 1.3 Structure of the article

The remaining structure is arranged as follows: Sect. 2 provides the overall framework of the proposed method. Section 3 elaborates the mathematical theory for the proposed method, including the SINDy, the sparse model selection with the Akaike information criterion (SINDy-AIC), and the group sparsity identification. Relevant results for experiments are displayed in Sect. 4. The discussion and further research are described in Sect. 5. Finally, the meaning of symbols appeared in this work is synthesized in “Appendix A”.

## 2 Framework of the proposed approach

In this work, the SINDy algorithm is extended to integrate with the Akaike information criterion (AIC) and the group sparsity to identify a class of typical ODE and PDE models. Noise with normal and uniform distributions is introduced to test the

robustness of the proposed approach. The framework and the flowchart of the suggested method in this paper are displayed in Figs. 1 and 2. The primary process is described as follows:

- (1) *Step 1: Sparse identification of nonlinear dynamics (SINDy).* A set of the time series data extracted from several ODE systems, prescribed as  $\mathbf{X}, \dot{\mathbf{X}}$ , is imported into the SINDy model and a predefined library  $\Theta(\mathbf{X})$  is constructed on the basis of a priori physical information. Next, the sparsity threshold  $\lambda$  is utilized to iteratively regularize the nonzero entries  $\xi_i$  in the matrix  $\Xi$  to obtain a sparse model  $\xi_i$ .
- (2) *Step 2: Sparse model selection.* The SINDy algorithm provides a combinatorially large amount of candidate models  $Model(j)$  in the training set  $\mathbf{X}$ . Then the Akaike information criterion (AIC) is incorporated into the SINDy to select the optimal model  $Model(inds(1))$  by hierarchically ranking the AIC scores  $IC(j)$  with the inclusion of different categories of support values in the validation set  $\mathbf{Y}$ .
- (3) *Step 3: Sequentially thresholded group ridge regression.* In the primitive SINDy architecture, the sequentially thresholded least squares is

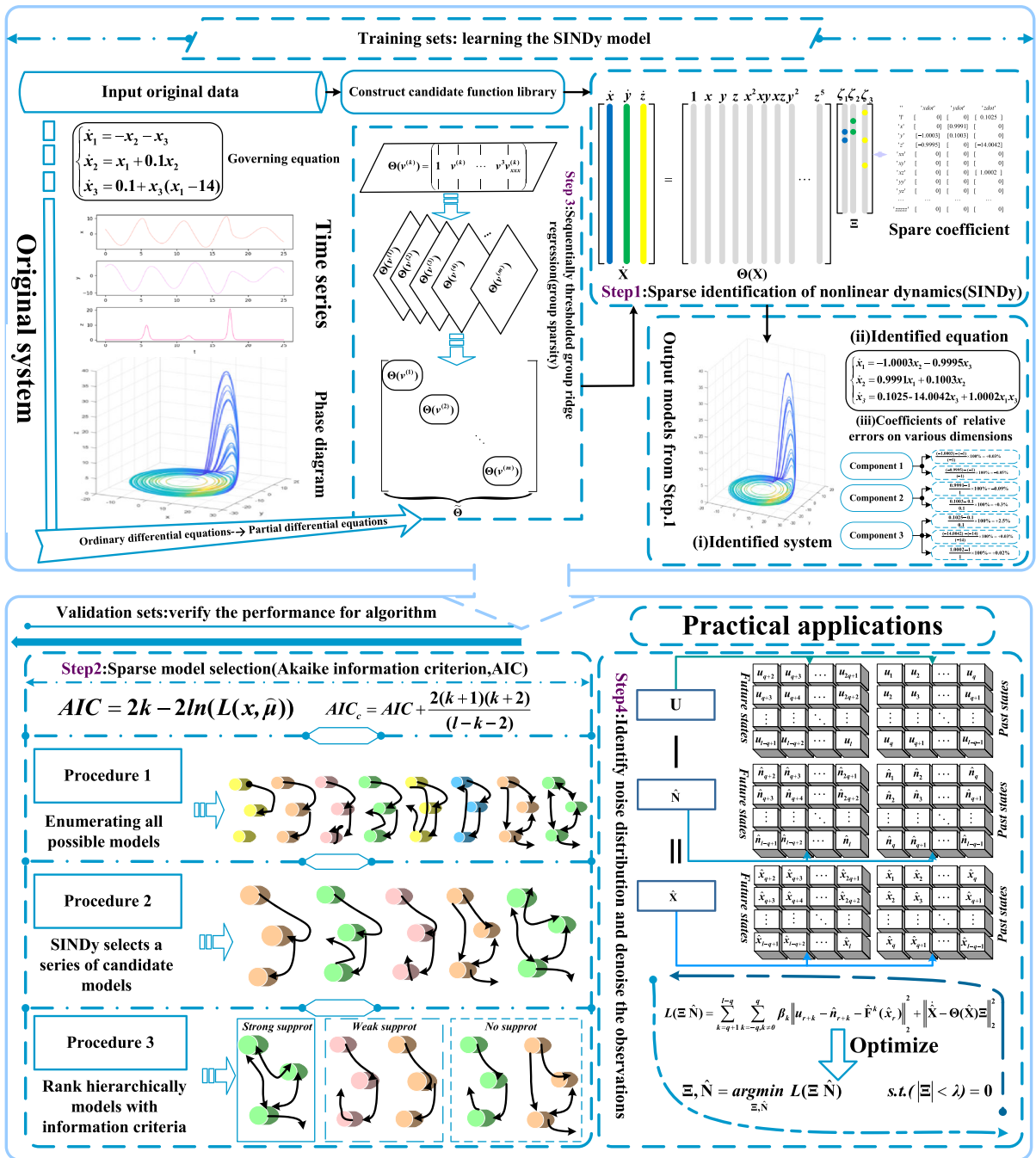


Fig. 1 Schematic of the proposed technique process

employed to learn the sparse terms  $\xi_l$  incorporated in each candidate function library  $\theta_l$ . Due to the complex structure of the parametric PDE dynamics, the group sparsity is used to learn the dependently parametric coefficients with

function libraries and sparse vectors bundled into a group  $\Gamma$ , where the ridge regression is leveraged to threshold the coefficients  $\xi_i$ .

(4) *Step 4: Automatic noise identification.* In general, the robustness of this approach should be

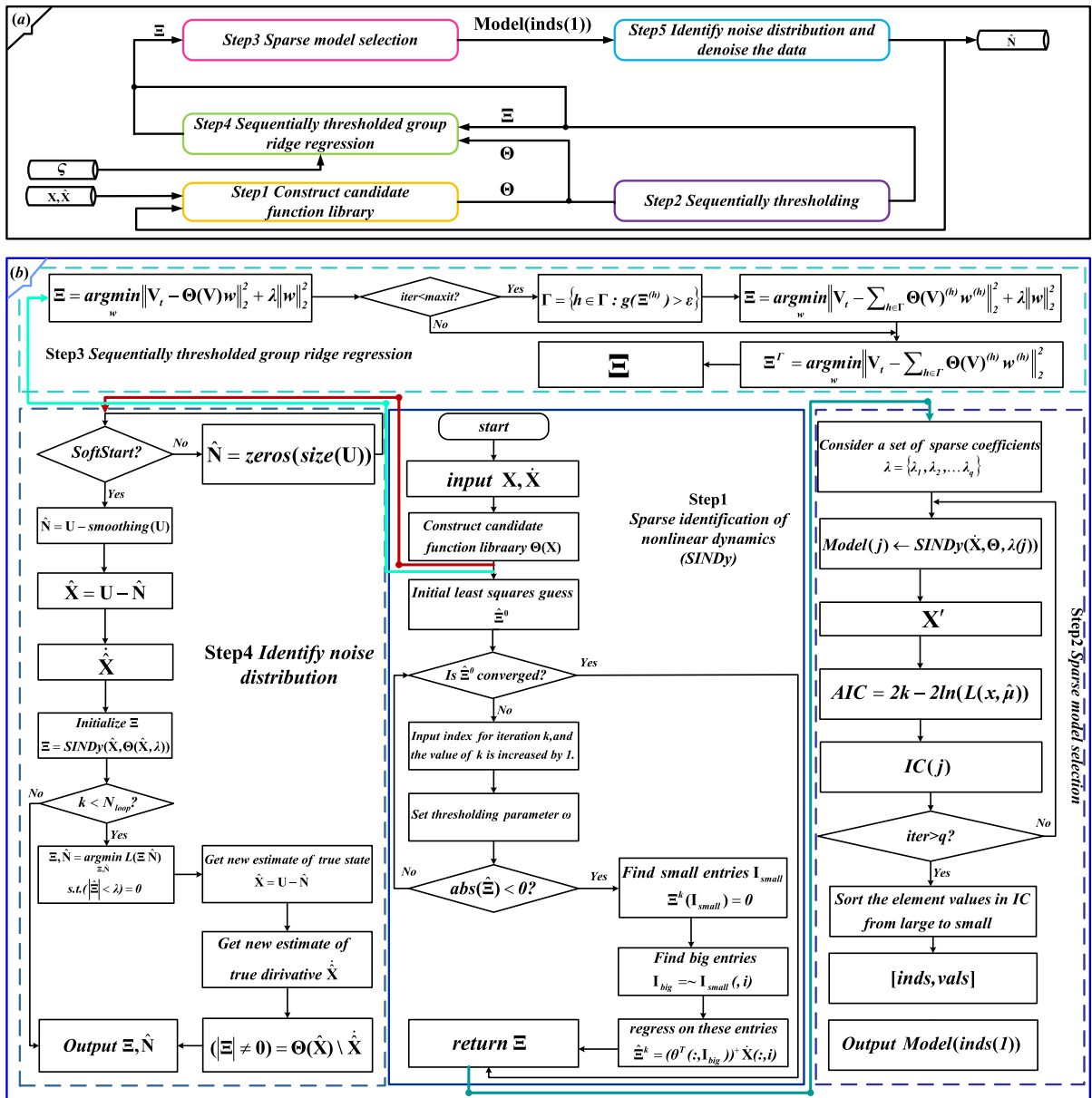


Fig. 2 Overall calculation flow of the adopted method

considered. First, the noise estimation  $\hat{N}$  is achieved by presmoothing the noisy observation  $U$  and obtaining the estimated value of the clean measurements  $\hat{X}$  and its derivative  $\dot{\hat{X}}$ . Finally, the technique makes use of optimization and the SINDy algorithm to synchronously recognize the distribution of the noise and to denoise the observations.

### 3 Methods

#### 3.1 SINDy

Here, we consider the form of equations for dynamical systems

$$\frac{d}{dt} \mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)) \tag{1}$$

**Table 2** The pseudo-code of Algorithm 1

<b>Algorithm 1 SINDy</b>	
<b>Inputs:</b> Observed measurements and time derivatives, $\mathbf{X}, \dot{\mathbf{X}} \in \mathbb{R}^{m \times n}$ dimension of state variables, $n$ , the utmost order of nonlinear terms contained in assuming right-hand-side of the equation, $p$ , sparsification knob, $\lambda$ .	
<b>Output:</b> Sparse vector of coefficients $\Xi$ .	
1: <b>procedure</b> SINDy( $\mathbf{X}, \dot{\mathbf{X}}$ )	
2: $\Theta \in \mathbb{R}^{m \times p} \leftarrow \text{poolData}(x, n, p)$	▶ Construct candidate function library
3: <b>for</b> $k \in \{1, 2, \dots, n\}$ <b>do</b>	▶ Different state variables
4: $\Xi_k = \Theta_k / \mathbf{X}_{t,k}$	▶ Initial guess: least-squares
5: <b>for</b> $l \in \{1, 2, \dots, 10\}$ <b>do</b>	▶ Sequentially thresholded
6: $\text{sinds} = (\text{abs}(\Xi_k) < \lambda)$	▶ Find small terms
7: $\Xi_k(\text{sinds}) = 0$	▶ Thresholding
8: $b_k \sim \text{sinds}(:, k)$	▶ Find big coefficients
9: $\Xi_k(b_k) = \Theta_k(b_k) / \mathbf{X}_{t,k}$	▶ Regress dynamics onto the rest terms to find $\Xi_k$
10: <b>end</b>	
11: $\Xi(:, k) = \Xi_k$	▶ Congress identified model coefficients
12: <b>end</b>	
13: <b>end procedure</b>	

The function  $\mathbf{f}(\mathbf{x}(t))$  denotes the governing equations of nonlinear dynamical systems, and the vector  $\mathbf{x}(t) \in \mathbb{R}^n$  indicates the shape of a system at time  $t$ .

Actually, there exist only a few important terms in governing equations for physical systems of interest, so that the right-hand side of the equations is assumed to be sparse in the space of potential functions. To identify the form of Equations  $\mathbf{f}$  from measurements, observations  $\mathbf{x}(t)$  extracted from dynamical systems in time history and sampled at several times  $t_1, t_2, \dots, t_m$  are arranged in the following matrix:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T(t_1) \\ \mathbf{x}^T(t_2) \\ \vdots \\ \mathbf{x}^T(t_m) \end{bmatrix} = \begin{bmatrix} x_1(t_1) & x_2(t_1) & \cdots & x_n(t_1) \\ x_1(t_2) & x_2(t_2) & \cdots & x_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ x_1(t_m) & x_2(t_m) & \cdots & x_n(t_m) \end{bmatrix} \tag{2}$$

However, as the observed data will be employed to learn a model that can capture the entire trajectory of the motion of systems, the amount of observations should not be extremely small. In this way, the data

can be augmented by coordination transformation or matrix transformation [59]. Here, the derivative  $\dot{\mathbf{x}}(t)$  calculated by numerically approximate or total variation regularization is applied as the augmentation of the data dimension, which can be represented by

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\mathbf{x}}^T(t_1) \\ \dot{\mathbf{x}}^T(t_2) \\ \vdots \\ \dot{\mathbf{x}}^T(t_m) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_2(t_1) & \cdots & \dot{x}_n(t_1) \\ \dot{x}_1(t_2) & \dot{x}_2(t_2) & \cdots & \dot{x}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_1(t_m) & \dot{x}_2(t_m) & \cdots & \dot{x}_n(t_m) \end{bmatrix} \tag{3}$$

Realistically,  $\mathbf{X}$  and  $\dot{\mathbf{X}}$  are often contaminated with noise. Relying on the noise, it may be indispensable to filter  $\mathbf{X}$  and  $\dot{\mathbf{X}}$ . Otherwise, the vector of sparse coefficients does not hold exactly. Total variation regularization [60] is utilized to denoise the derivative  $\dot{\mathbf{X}}$  to counteract the differentiation error.

As control variables constituting dynamics can be established by individual customs, it is promising to make an assumption that the control variables can be deemed known terms. Accordingly, a candidate



function, which includes all the possible nonlinearities composed of control variables in the form of permutations, including constant, polynomial or trigonometric terms, can be constructed. The specific form can be defined by the form of the governing equations of nonlinear dynamics,

$$\Theta(\mathbf{X}) = \begin{bmatrix} | & | & | & | & \cdots & | & | & | \\ 1 & \mathbf{X} & \mathbf{X}^{Q_2} & \mathbf{X}^{Q_3} & \cdots & \sin(\mathbf{X}) & \cos(\mathbf{X}) & \cdots \\ | & | & | & | & & | & | & | \end{bmatrix} \quad (4)$$

where  $\mathbf{X}^{Q_2}, \mathbf{X}^{Q_3}$  denotes a higher order polynomial term. For instance,  $\mathbf{X}^{Q_2}$  represents the quadratic nonlinearities with respect to the state  $\mathbf{x}$ , the exact form of which is shown as

$$\mathbf{X}^{Q_2} = \begin{bmatrix} x_1^2(t_1) & x_1(t_1)x_2(t_1) & \cdots & x_2^2(t_1) & \cdots & x_n^2(t_1) \\ x_1^2(t_2) & x_1(t_2)x_2(t_2) & \cdots & x_2^2(t_2) & \cdots & x_n^2(t_2) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_1^2(t_m) & x_1(t_m)x_2(t_m) & \cdots & x_2^2(t_m) & \cdots & x_n^2(t_m) \end{bmatrix} \quad (5)$$

At this point, Eq. (1) can be transformed into the form of Eq. (6). In this case, it is assumed that the left-hand side of the equation and the first term on the right-hand side of the equation represented as a self-set library are known, and the quantity to be obtained is the matrix of coefficients that characterize the control terms  $\Xi = [\xi_1 \ \xi_2 \ \cdots \ \xi_n]$ .

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi \quad (6)$$

Simply put, the problem for obtaining specific governing equations is converted to a requirement for the vector of sparse coefficients. Naturally, symbolic regression is exploited to simplify the issue. Therefore, Eq. (6) is substituted with Eq. (7).

$$\dot{\mathbf{x}}_l = \mathbf{f}_l(\mathbf{x}) = \Theta(\mathbf{x}^T)\xi_l \quad (7)$$

Here,  $\Theta(\mathbf{x}^T)$ , which is a numerical matrix, is a vector of the symbolic function of elements of  $\mathbf{x}$  with respect to  $\Theta(\mathbf{X})$ . Naturally, the final form of the equations to be solved is indicated by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) = \Xi^T(\Theta(\mathbf{x}^T))^T \quad (8)$$

where the superscript  $T$  denotes the matrix transpose.

Each column of Eq. (6) demands a different optimization to obtain the sparse vector of coefficients  $\xi_l$  for the  $l$ th row equation. To mitigate computational errors caused by inconsistent data dimensions or

excessively small entries of  $\mathbf{X}$ , columns of  $\Theta(\mathbf{X})$  may be normalized. In view of presupposing that most coefficient matrices are sparse in an appropriate basis, a sparse solution to an overdetermined system with noise should be sought. Here, the sequentially thresholded least squares [24] is used, the particular form of which is given by

$$\min \frac{1}{2} \|\dot{\mathbf{X}} - \Theta(\mathbf{X})\Xi\|^2 + \lambda R(\Xi) \quad (9)$$

where  $\lambda$  is the sparsity knob and the  $R(\cdot)$  denotes the regularized function.

Once the coefficient matrix has been acquired, the identified model through SINDy is obtained, which can also be considered a reconstruction or prediction of the ground truth system. The detailed flow is shown in Figs. 1 and 2, and Algorithm 1 outlines its pseudocode in Table 2.

### 3.2 Sparse model selection (SINDy-AIC)

For higher accuracy, the models obtained by regression analysis may be further refined in conjunction with statistical learning methods. There are many model selection methods, including the Akaike information criterion (AIC) [19, 20], Bayesian information criterion (BIC) [21], deviance information criterion (DIC) [23] and cross-validation (CV) [22].

In this paper, the AIC is chosen as the statistical score for model comparison from many combinatorially probable models. Originally pioneered by Akaike.H. [19, 20], AIC incorporated the principles of information entropy and Kullback–Leibler (K-L) distance [18] and built upon the notion of maximum likelihood estimation to appraise parameters. Given a candidate *Model*  $i$ , its corresponding AIC value is

$$AIC_i = 2k - 2 \ln(L(\mathbf{x}, \hat{\mu})) \quad (10)$$

where  $L(\mathbf{x}, \hat{\mu}) = P(\mathbf{x}|\hat{\mu})$  denotes the conditional probability function for state variable  $\mathbf{x}$  under the condition of the estimation  $\hat{\mu}$ , and  $k$  indicates the number of free parameters. It should be noted that  $2k$  represents a penalty enforced on the lower bound of the AIC values. In a wide variety of cases, the sampled data are finite datasets and a correction to the AIC value is required by

$$AIC_c = AIC + \frac{2(k+1)(k+2)}{(l-k-2)} \quad (11)$$



where  $l$  refers to the dimension of the observed data. In linear regression, the residual sum of squares (RSS) is generally applied to least-squares fitting for the objective function, which is embodied in the form of  $RSS = \sum_{j=1}^l (y_j - g(x_j; \mu))^2$ , where  $x_j$  denotes the independent variable for observation,  $y_j$  signifies the observed dependent variable, and the candidate model is defined by  $g$ . Thus, Eq. (10) can be reformulated as  $AIC = l \ln(RSS/l) + 2k$  and its value will be used in Eq. (11).

Indeed, the AIC scores prefer a rescaled criterion named by  $AIC_{min}$  with the value of the minimum AIC since there are multiple AIC values for each potential models. In addition, the relative AICc can be linked to the statistical indicator  $p$ -value, so that the relative AIC values  $\Delta_i = AIC_i - AIC_{min}$  can be straightforwardly defined as a strength-of-evidence criterion for a contrast with informative models [61]. The strong support value corresponds to the models with  $\Delta_i \leq 2$ , and the weak support is relative to the models with  $4 \leq \Delta_i \leq 7$ , while the models with  $\Delta_i \geq 10$  are interpreted as no support.

The observations are divided into a training set and a validation set to verify the performance of this approach. The length and the sampling frequency of the time-series traces in each validation set are the same as the homologous training data, except for the initial conditions. For each example in Sect. 4.2, to validate the noise sensitivity, Gaussian noise with mean zero and standard deviation  $\varepsilon = 1.5 \times 10^{-4}$  is added to both the training sets and the validation sets. The sparse threshold  $\lambda$  appearing in the validation sets is optimized by cross-validation.

### 3.3 Identification of parametric PDEs with group sparsity

In this section, the partial differential equation functional identification of nonlinear dynamics (PDE-FIND) [62] is introduced firstly. The mathematical form of the PDEs with constant coefficients is formulated as

$$v_t = N(v, v_x, v_{xx}, \dots) = \sum_{k=1}^q N_k(v, v_x, v_{xx}, \dots) \xi_k \quad (12)$$

where the unknown nonlinear function  $N(\cdot)$  is assumed as the summation of monomial basis functions  $N_k$

comprised by variable  $v$  and its differential values, and the subscripts indicate the partial equation.

For simplicity, identifying a constant coefficient PDE can be idealized as an overdetermined linear regression problem  $\mathbf{V}_t = \Theta(\mathbf{V})\xi$ , its detailed formulation is illustrated by

$$\underbrace{\begin{bmatrix} v_t(x_1, t_1) \\ v_t(x_2, t_1) \\ \vdots \\ v_t(x_n, t_m) \end{bmatrix}}_{\mathbf{V}_t} = \underbrace{\begin{bmatrix} 1 & v(x_1, t_1) & v_x(x_1, t_1) & \dots & v^3 v_{xxx}(x_1, t_1) \\ 1 & v(x_2, t_1) & v_x(x_2, t_1) & \dots & v^3 v_{xxx}(x_2, t_1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & v(x_n, t_m) & v_x(x_n, t_m) & \dots & v^3 v_{xxx}(x_n, t_m) \end{bmatrix}}_{\Theta(\mathbf{V})} \xi \quad (13)$$

Note that measurement matrix  $\mathbf{V}_t \in \mathbb{R}^{m \times n}$  demonstrates data sampled on  $m$  time instances and  $n$  spatial grids.  $\Theta(\mathbf{V})$  is a matrix composed with potential models and  $\xi$  represents the sparse vector of coefficients.

In contrast to PDE-FIND, the coefficients of either temporally or spatially parameter-dependent PDEs are required to be learned in this work. The time-varying parametric dependent expressions are shown as

$$v_t = N(v, v_x, v_{xx}, \dots, \mu(t)) = \sum_{k=1}^Q N_k(v, v_x, v_{xx}, \dots) \xi_k(t) \quad (14)$$

The form of Eq. (14) has similarity to Eq. (12) except for the parametric term  $\mu(t)$ . If the coefficient estimates are obtained in variable space,  $\xi(x)$  is selected to substitute  $\xi(t)$ .

Similar to Eq. (4), the guessing candidate function library is constructed as

$$\Theta(v^{(k)}) = \begin{pmatrix} | & | & | & | \\ 1 & v^{(k)} & \dots & v^3 v_{xxx}^{(k)} \\ | & | & | & | \end{pmatrix} \quad (15)$$

It encapsulates cross terms constituted by derivatives and parameters, where the form of the series of  $m$  equations is shown by Eq. (16).

$$v_t^{(k)} = \Theta(v^{(k)}) \xi^{(k)}, \quad k = 1, \dots, m \quad (16)$$

Unlike the sparse regression problems in SINDy, the constraint that all the  $\xi^{(k)}$  share the same sparsity threshold is considered in this article. Therefore, the notion of group sparsity [63] is introduced to group sets of features in  $\Theta(v^{(k)})$  together. Then Eq. (13) can be transformed into the block diagonal form

$$\underbrace{\begin{bmatrix} v_t^{(1)} \\ v_t^{(2)} \\ \vdots \\ v_t^{(m)} \end{bmatrix}}_{\mathbf{v}_t} = \underbrace{\begin{bmatrix} \Theta(v^{(1)}) & & & \\ & \Theta(v^{(2)}) & & \\ & & \ddots & \\ & & & \Theta(v^{(m)}) \end{bmatrix}}_{\Theta(\mathbf{v})} \begin{bmatrix} \xi^{(1)} \\ \xi^{(2)} \\ \vdots \\ \xi^{(m)} \end{bmatrix} \tag{17}$$

Similar to Eq. (13), the class of equations represented by Eq. (16) can also be regarded as a solitary linear system. And the group sparsity will be involved in the original SINDy algorithm, then Eq. (9) will change to

$$\hat{\xi} = \underset{\xi}{\operatorname{argmin}} \left\| \mathbf{V}_t - \sum_{h \in \Gamma} \Theta(\mathbf{V})^{(h)} \xi^{(h)} \right\|_2^2 + \lambda \|\xi\|_2^2 \tag{18}$$

where  $\Gamma$  denotes the groups, for  $m$  time grids and  $q$  potential features in library,  $\Gamma$  is defined as  $\Gamma = \{k + q \cdot i : i = 1, \dots, m : k = 1, \dots, q\}$ . And  $\tilde{\Theta}(\mathbf{V})$  indicates the block diagonal matrix  $\Theta(\mathbf{V})$ .

Indeed, the approach evenly divides the measurements into different time steps or spatial locations with respect to the varied coefficients. Additionally, the validation set is constructed with 20% sampled data. And 20 cross-validation experiments are used to get the optimal solution to the sparse threshold. Next the well-selected models will be evaluated with the AIC-enlightened loss function:

$$L(\xi) = n \ln \left( \frac{\|\tilde{\Theta} \xi - \tilde{v}_t\|_2^2}{n} + \lambda_f \right) + 2k \tag{19}$$

where  $n$  represents the amount of rows in  $\Theta$ ,  $k$  denotes the amount of the non-zero elements in the learned PDEs,  $\lambda_f$  indicates the threshold benchmark that exerts on each model to prevent over-fitting, and  $\tilde{v}_t$  represents the normalized vector of  $v_t$  that all terms in it have unit length.

### 3.4 Identify noise

In practice, a wide range of systems of interest can be affected by many adverse factors such as noise, which are usually ineradicable. Indeed, the challenging problems in actual application are how to identify the regime of the noise distribution and how to learn the type of noise apart from the Gaussian noise with mean zero and standard variance  $\varepsilon = 1$ .

Crucially, noise can be considered as additional data added to the clean data [64]. Based on this notion, the observations  $\mathbf{u}(t) = [u_1(t), u_2(t), \dots, u_p(t)] \in \mathbb{R}^{1 \times p}$  can be classified into noiseless data  $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_p(t)] \in \mathbb{R}^{1 \times p}$  and noisy estimation  $\mathbf{n}(t) = [n_1(t), n_2(t), \dots, n_p(t)] \in \mathbb{R}^{1 \times p}$ .

$$\mathbf{u}(t) = \mathbf{x}(t) + \mathbf{n}(t) \tag{20}$$

Therefore, the candidate function library can also consist of two parts: the noiseless measurements  $\mathbf{X} = [\mathbf{x}(t_1); \mathbf{x}(t_2); \dots; \mathbf{x}(t_l)] \in \mathbb{R}^{l \times p}$  and the noisy measurements  $\mathbf{N} = [\mathbf{n}(t_1); \mathbf{n}(t_2); \dots; \mathbf{n}(t_l)] \in \mathbb{R}^{l \times p}$ , which is illustrated as

$$\dot{\mathbf{U}} = \dot{\mathbf{X}} + \dot{\mathbf{N}} = \Theta(\mathbf{U})\Xi = \Theta(\mathbf{X} + \mathbf{N})\Xi \tag{21}$$

Throughout the training, the advent of noise has the implication on the correctness of the model identification. Therefore, it is essential to isolate the noise from signals as two segments, the noiseless observations and the noisy observations, whose patterns are constituted with past states and future states, in which step  $q$  denotes the degree advanced to current snapshots. Accordingly, the model is illustrated in integration form:

$$\mathbf{x}(r + q) = \mathbf{F}^q(\mathbf{x}(r)) = \mathbf{x}(r) + \int_{t_r}^{t_{r+q}} \Theta(\mathbf{x}(\tau))\Xi d\tau \tag{22}$$

where  $\Theta(\mathbf{x}(\tau))\Xi$  indicates the vector field of the system, and  $\mathbf{F}(\cdot)$  denotes the nonlinear dynamics.

The two parts are then distinctly solved for sparse coefficients, and the result is compromised with the summation of distinct products. Aiming to diminish the absolute error between the estimated derivative  $\dot{\hat{\mathbf{X}}}$  and the actual derivative  $\Theta(\hat{\mathbf{X}})\Xi$ , named,  $e_d$ , which is shown as

$$e_d = \left\| \dot{\hat{\mathbf{X}}} - \Theta(\hat{\mathbf{X}})\Xi \right\|_2^2 \tag{23}$$

The additional constraint would be needed to regularize the Eq. (23) to couple the optimization parameters  $\hat{\mathbf{N}}$  and  $\Xi$ . Therefore, the formulation is revised by

$$e_{s,r} = \sum_{l=-q, l \neq 0}^q \omega_l \left\| \mathbf{u}(r+l) - \hat{\mathbf{n}}(r+l) - \hat{\mathbf{F}}^l(\hat{\mathbf{x}}(r)) \right\|_2^2 \tag{24}$$

where  $\omega_l$  is used to constrain the numerical error,  $h$  is the decay factor, which is set to 0.9, and the details can be described as  $\omega_l = h^{|l|-1}$  ( $0 \leq h \leq 1$ ). Particularly, upper triangle symbols represent the estimation value and the subscripts indicate the time step.

As the validation set is separated for noiseless and noisy estimation, the loss function is naturally composed of two elements: numerical error  $e_d$  and simulation error  $e_s$ . Additionally,  $e_s$  is provided by the simulation of the vector field across the overall trajectory.

$$\begin{aligned}
 e_s &= \sum_{r=q+1}^{k-q} e_{s,r} \\
 &= \sum_{r=q+1}^{k-q} \sum_{l=-q, l \neq 0}^q \omega_l \left\| \mathbf{u}(r+l) - \hat{\mathbf{n}}(r+l) - \hat{\mathbf{F}}^l(\hat{\mathbf{x}}(r)) \right\|_2^2
 \end{aligned}
 \tag{25}$$

$$\begin{aligned}
 L(\Xi, \hat{\mathbf{N}}) &= e_s + e_d \\
 &= \sum_{r=q+1}^{k-q} \sum_{l=-q, l \neq 0}^q \omega_k \left\| \mathbf{u}(r+l) - \hat{\mathbf{n}}(r+l) - \hat{\mathbf{F}}^l(\hat{\mathbf{x}}(r)) \right\|_2^2 + \left\| \dot{\hat{\mathbf{X}}} - \Theta(\hat{\mathbf{X}})\Xi \right\|_2^2
 \end{aligned}
 \tag{26}$$

Since certain errors are bound to exist due to the extra noise, the same performance criterion advocated by Rudy et al. [65] is applied to judge the performance of the model. Specifically, these errors are the noise estimation error  $E_N$ , the vector field error  $E_f$  and the simulated trajectory error  $E_P$ . Notably, the relative noise estimation error is

$$E_N = \frac{1}{k} \sum_{l=1}^k \left\| \mathbf{n}_l - \hat{\mathbf{n}}_l \right\|_2^2
 \tag{27}$$

which is defined by the inconsistency between the true noise  $\mathbf{n}_l$  and the identified noise  $\hat{\mathbf{n}}_l$  with the mean  $\ell_2$  error. The vector field error is

$$E_f = \frac{\sum_{l=1}^k \left\| \mathbf{f}(\mathbf{x}_l) - \hat{\mathbf{f}}(\mathbf{x}_l) \right\|_2^2}{\sum_{l=1}^k \left\| \mathbf{f}(\mathbf{x}_l) \right\|_2^2}
 \tag{28}$$

which denotes the deviation from the identified vector field  $\hat{\mathbf{f}}(\mathbf{x}_l)$  to the true vector field  $\mathbf{f}(\mathbf{x}_l)$  with relative squared  $\ell_2$  error. The simulated trajectory error is

$$E_P = \frac{1}{\|\mathbf{X}\|_F^2} \sum_{k=1}^{l-1} \left\| \mathbf{x}_l - \hat{\mathbf{F}}^l(\mathbf{x}_l) \right\|_2^2
 \tag{29}$$

which represents the discrepancy between the true trajectory  $\mathbf{x}_l$  and the forward simulation trajectory

$\hat{\mathbf{F}}^l(\mathbf{x}_l)$ , especially,  $\|\cdot\|_F$  denotes the Frobenius norm. The exact process is shown in Figs. 1 and 2.

## 4 Results

In this section, we demonstrate the effectiveness of the above methods by several hybrid dynamical systems: the *Van der Pol system*, the *Rössler system*, the temporally dependent *Burgers' equation* and the spatially dependent *Kuramoto–Sivashinsky equation*, respectively. Finally, a class of physical systems are provided to investigate the robustness of the proposed approach here.

### 4.1 Experiment I: Sparse identification of nonlinear dynamics

#### 4.1.1 Van der Pol system

The *Van der Pol system* [66] is defined by

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = 5(1 - x_1^2)x_2 - x_1 \end{cases}
 \tag{30}$$

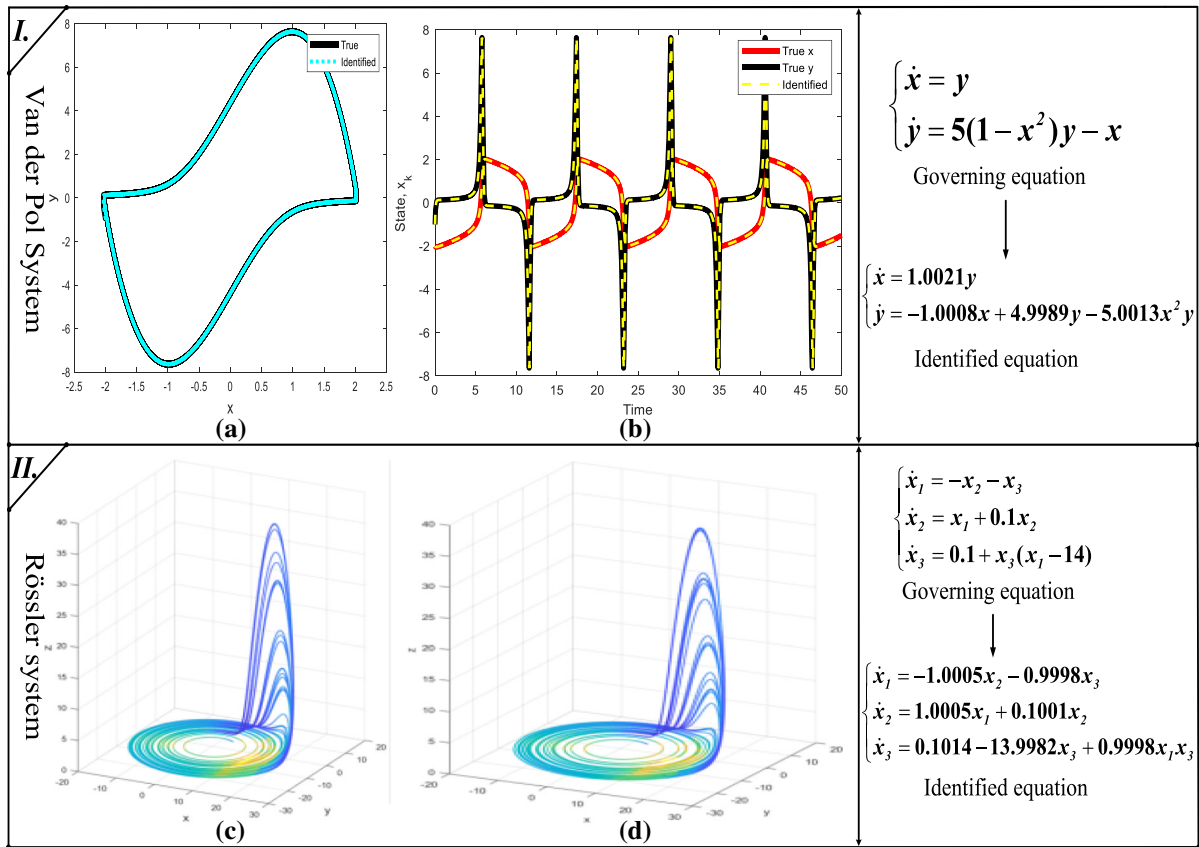
The initial condition is given by  $[x_{10}, x_{20}] = [-2, -1]$  and the training set is acquired at interval  $[-25, 25]$  as the snapshot  $t = 0.001$ .

Figure 3a shows a comparison of the actual trajectory with the identified trajectory for the *Van der Pol system* in the form of a phase diagram, where it shows that the learned trajectory is approximated to the true trajectory with the extremely small error. Figure 3b decomposes the two-dimensional phase into two dimensions:  $x$  and  $y$ . Similarly, the result of the comparison between the actual time-series data and the estimated time-series instances on two dimensions shows that the error between them is extremely small, even almost zero.

#### 4.1.2 Rössler system

The *Rössler system* [67] is also used to verify the performance of SINDY, which is governed by

$$\begin{cases} \dot{x}_1 = -x_2 - x_3 \\ \dot{x}_2 = x_1 + 0.1x_2 \\ \dot{x}_3 = 0.1 + x_3(x_1 - 14) \end{cases}
 \tag{31}$$



**Fig. 3** Example of nonlinear dynamics to test the performance of the SINDy algorithm

This system is simulated with initial condition  $[x_{10}, x_{20}, x_{30}] = [3, 5, 0]$ . The training set is sampled with the interval  $t = 0.001$  from  $t = 0$  to  $t = 25$ .

Figure 3c represents the actual *Rössler system*, while Fig. 3d is the identified *Rössler system* via SINDy. Comparing the two panels, it is known that SINDy accurately reconstructs the *Rössler system*. Simultaneously, the results of the coefficients indicate that the model error between them is within the range from 0.01% to 0.05%.

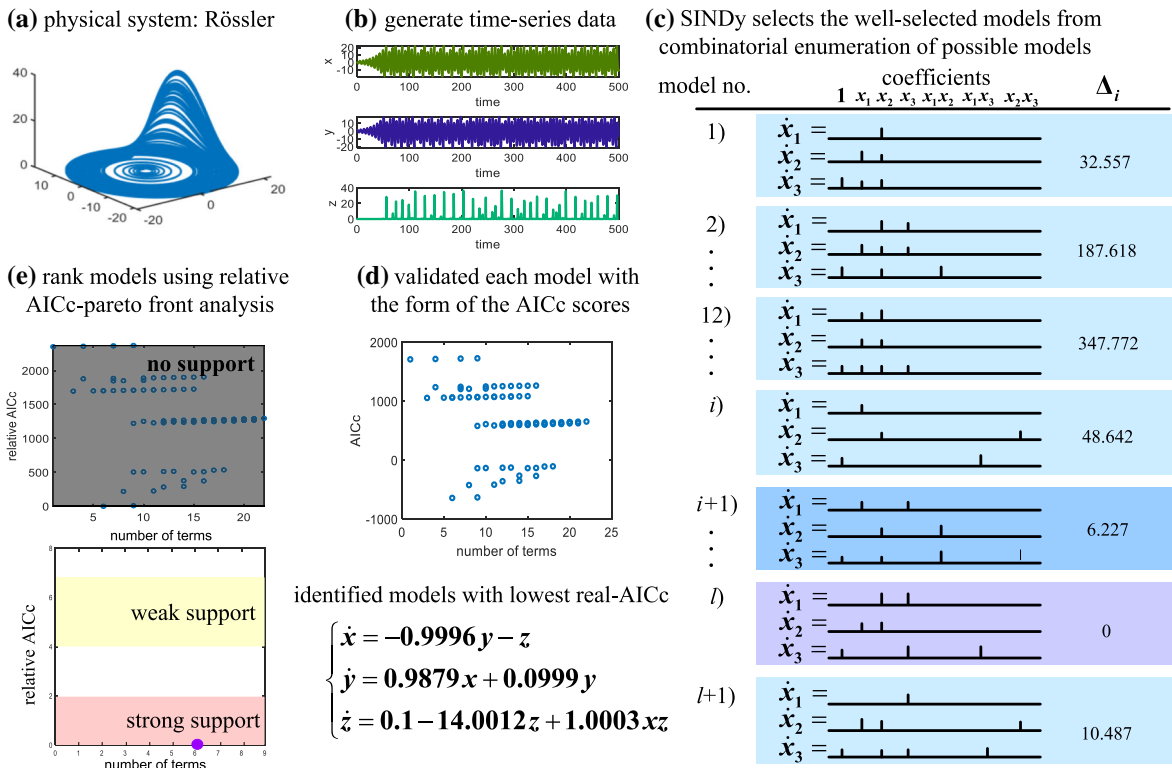
## 4.2 Experiment II: Sparse model selection

### 4.2.1 Rössler system

The *Rössler system* includes three state variable ( $m = 3$ ) with a third-order polynomial library ( $p = 3$ ). The total number of models is  $N_c =$

$\sum_{i=1}^{N_a} \binom{N_a}{i} = 1023$  in the library, each of them is denoted with  $N_a = \binom{m+p}{p} = 10$  possible monomials. One hundred initial conditions are randomly given to generate distinct validation sets with identical dimensions as the training sets, so that 100 cross-validation experiments are enforced on the validation set to adjust the values of the hyper-parameter  $\lambda$  and to determine the optimal model to be estimated.

As shown in Fig. 4c–e, there are so sizeable candidate models selected from the hierarchically ranked relative AICc that the rescaled relative AICc is used to narrow the range of candidate models. It is found that at the position relevant to the value of 6, the value of the relative AICc falls dramatically. Simultaneously, as shown in Fig. 4e, the 6-term model is equipped with ‘strong support’. Therefore, the model involving 6 nonlinearities is the optimal model.



**Fig. 4** Selected sparse models for the *Rössler system* with SINDy-AIC

### 4.2.2 Van der Pol oscillator

The *Van der Pol system* is taken for the test case, whose formulation is governed by Eq. (30), with two state variables ( $m = 2$ ) and a sixth-order polynomial library ( $p = 6$ ). On this occasion, there exist  $N_a = \binom{m+p}{p} = 28$  potential monomials, and  $N_c = \sum_{i=1}^{N_a} \binom{N_a}{i} \approx 2.68435 \times 10^8$  possible models. One hundred cross-validation trials are performed for each system on a validation set of the same dimension as the training set for ranking AICc.

Only the individual trajectory for each state variable ( $x$  and  $y$ ) of the *Van der Pol* is employed as input, as shown in Fig. 5a. SINDy is able to subselect a series of models from the candidate library, including models with 1, 2 and 4 terms, as shown in Fig. 5b. Notably, there is an abrupt drop in the value of relative AICc at the number of nonlinear terms = 4, and the support values of the selected models fall into the

strongly supported range, while other models fall into the unsupported domain according to Fig. 5b. Therefore, the optimal model selected is the one containing four nonlinear terms.

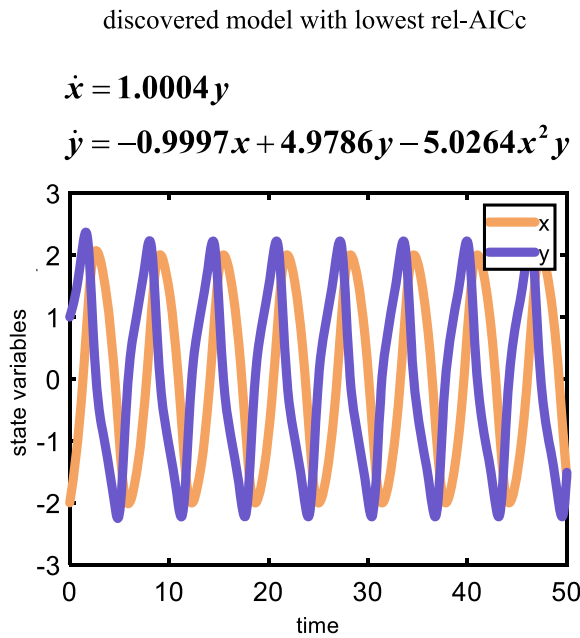
Finally, the value of the prediction error of SINDy-AIC is less than that of SINDy within the range of noise level from  $10^{-8}$  to 0.5, as shown in Fig. 5c. Nevertheless, the larger the noise value is, the greater the prediction of SINDy-AIC, with a value close to 1. The robustness of SINDy-AIC can perhaps be the focus of future research.

### 4.3 Experiment III: Identify the coefficients of the partial differential equations

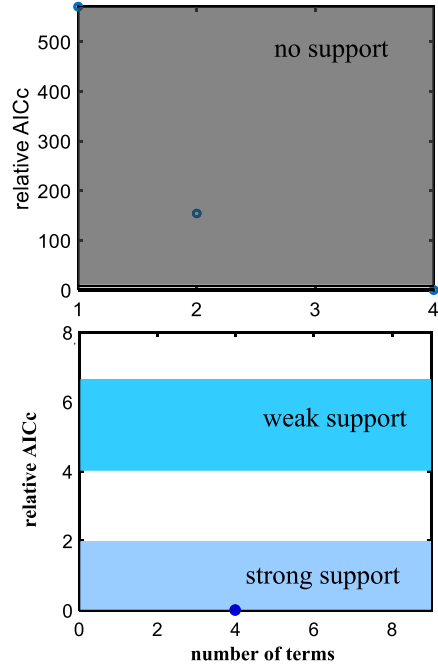
#### 4.3.1 Burgers' equation

The first example is focused on *Burgers' equation*, which has a nonlinear advection term in the form of a sinusoidally oscillatory coefficient  $\omega(t)$ , which is defined by

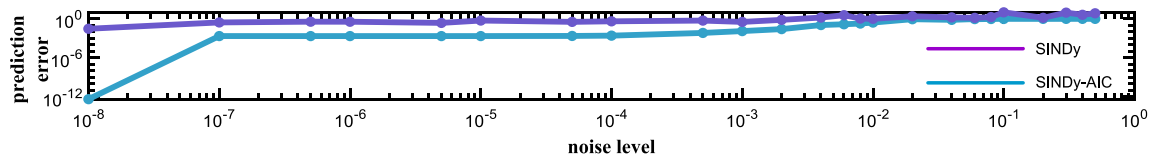
(a) time-series data of the van der pol system



(b) analyze well-selected models with the relative-AIC



(c) comparison of SINDy and SINDy-AIC given by the van der pol system



**Fig. 5** Selected models for the *Van der Pol* system with SINDy-AIC

$$v_t = \omega(t)vv_x + 0.1v_{xx}$$

$$\omega(t) = -\left(1 + \frac{\sin(t)}{4}\right) \tag{32}$$

Data are generated on the interval  $[-8, 8]$  via the spectral method and the periodic boundary requirements are constrained by  $k = 256$  snapshots and  $l = 256$  time instances to evenly segment the time scale and to construct the time-varying dependent parametric PDE. Our library of candidate functions is fabricated by employing the cubic order derivatives of  $v$ , which can be multiplied by the fourth-order derivatives of  $v$ . It is necessary to segregate 20% of the data from each snapshot to be utilized as a validation set. The proper level of sparsity requires the optimal  $\lambda$  through cross-validation to build the sparse coefficients.

The consequent time series for the identified coefficient  $\tilde{\omega}(t)$  and the true coefficient  $\omega(t)$  in the

case of either noiseless data or noisy data are illustrated in Fig. 6c, d, respectively. The results show that the parametric coefficients are identified elaborately with subtle error.

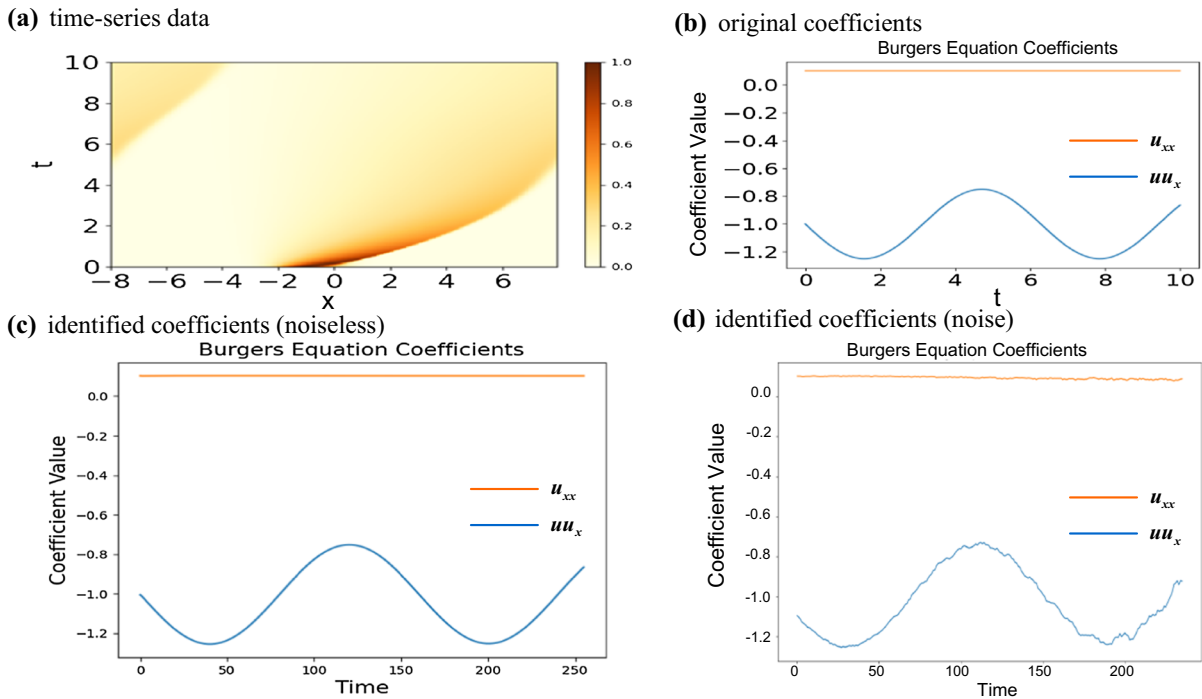
#### 4.3.2 Kuramoto–Sivashinsky equation

The *Kuramoto–Sivashinsky equation* is a nonlinear PDE that simulates it with spatially varying coefficients

$$v_t = \omega(x)vv_x + \zeta(x)v_{xx} + \gamma(x)v_{xxx} \tag{33}$$

where  $\omega(x) = 1 + \sin(\frac{\pi}{2}x/L)/5$ ,  $\zeta(x) = -1 + e^{-(x-3)^2/4}/5$ ,  $\gamma(x) = -1 - e^{-(x+3)^2/4}/5$  and  $L = 20$ . The spatially dependent *Kuramoto–Sivashinsky equation* is solved mathematically employing a spectral method on a periodic domain  $[-20, 20]$  with  $k = 512$  snapshots and  $l = 512$  time instances. The library





**Fig. 6** Identified coefficients of parametric *Burgers' equation* with or without the introduction of noise

consists of the product between the derivatives of  $v$  and its cubic order terms up to fourth order.

Figure 7b–d show the differences between the true coefficients and the learned coefficients in the noise-free and noisy measurements, respectively. Traditionally, the goodness-of-fit of the model is judged by the mean square error (MSE)  $\|\tilde{\Theta}\xi - \tilde{v}_t\|_2^2/n$  in Eq. (19). And the related mean square errors in two cases are  $3.92465 \times 10^{-6}$  and  $8.54312 \times 10^{-3}$ . Obviously, this method can correctly determine the coefficients for Kuramoto–Sivashinsky in the former, while it does not accurately learn the coefficients with the introduce of noise.

Additionally, the AIC can be employed to evaluate the discrepancy for true coefficients and identified coefficients, as shown in Fig. 7e. When the noise magnitude  $\varepsilon$  is less than 1, the differences between the identified model and the true model is close to 0.

In particular, the range of the threshold tolerances, from  $\kappa_{\min}$  to  $\kappa_{\max}$ , is initially employed to select the potential models, where  $\kappa_{\min}$  demonstrates the minimum tolerance that has consequences for the sparsity of the  $\|\Theta(\mathbf{V}_t)\xi - \mathbf{V}_t\|$ , and  $\kappa_{\max}$  indicates the

minimum tolerance that ensures all coefficients in  $\xi$  to be zero. And  $\kappa_{\min}/\kappa_{\max}$  is illustrated as  $\kappa_{\min}/\kappa_{\max} = \kappa_{\min}/\kappa_{\max} \|\xi_{\text{ridge}}\|_2$ , where  $\xi_{\text{ridge}} = (\tilde{\Theta}^T \tilde{\Theta} + \lambda \mathbf{I})^{-1} \tilde{\Theta}^T \tilde{v}_t$ .

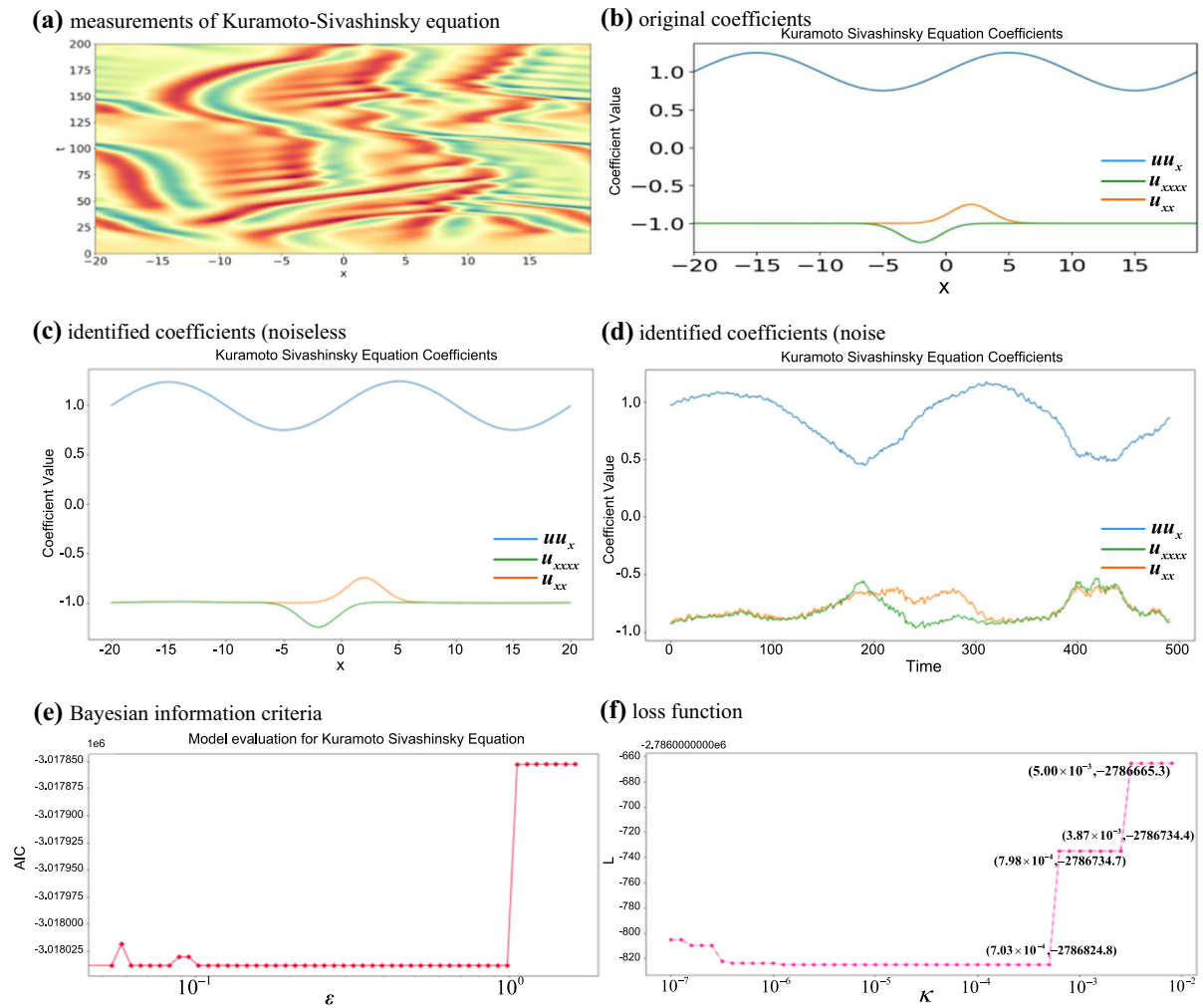
The notion of  $\tilde{\Theta}$  and  $\tilde{v}_t$  are illustrated in Sect. 3.3,  $\lambda$  is the threshold, and  $\mathbf{I}$  indicates the identity matrix.

Figure 7f provides the model evaluation with the loss function from Eq. (19). From 50 values of  $\kappa$ , ranging from  $10^{-7}$  to  $10^{-2}$  evenly spaced in steps of  $10^{-1}$ , the candidate models will be selected in both intervals of  $\kappa$  from  $7.03 \times 10^{-4}$  to  $7.98 \times 10^{-4}$  and from  $3.87 \times 10^{-3}$  to  $5.00 \times 10^{-3}$ . Furthermore, the models corresponding to  $\kappa_{\min}$  within an interval are overfit to the measurements, while the models related to  $\kappa_{\max}$  are too sparse to predict the coefficients. Then the AIC will be enforced to yield the optimal model for the PDEs.

### 4.3.3 Comparison results

Regarding the validity of the augmented SINDy with group sparsity, the structure error and the success rate are introduced. The amount of entries in the model that





**Fig. 7** Identified coefficients of the spatial Kuramoto–Sivashinsky equation and its respective performance indices

are abnormally increased or removed are denoted by the structure error. As seen in Fig. 8a, the indigo area implies no discrepancy between the discovered model and the actual model, and the violin patterns quantify the allocation of inaccurate entries with cross-validation among 20 models. It is obvious that the augmented SINDy with group sparsity can handle 30,000 more noise data than SINDy-AIC.

Figure 8b shows the comparison of the success rate of SINDy, SINDy-AIC and the augmented SINDy with group sparsity in discovering the coefficients of Burgers’ equation, where the success rate is denoted by the average of the results of 15 runs. The augmented SINDy employs approximately 5 times less data than SINDy-AIC, and SINDy-AIC utilizes

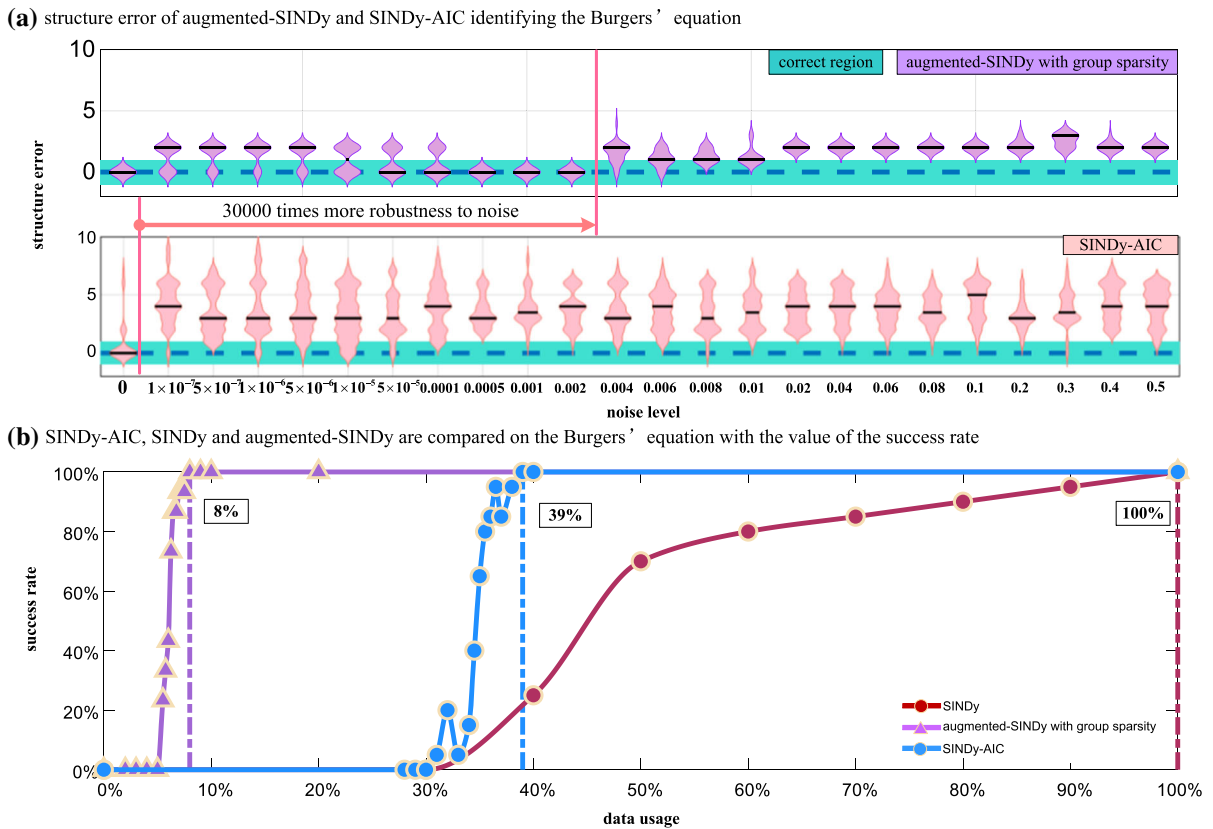
approximately 2 times less data than SINDy across a distinct percentage of measurements.

#### 4.4 Experiment IV: Noise identification

##### 4.4.1 Van der Pol System

Notably, the Van der Pol system is taken as the test instance, and its equation is formulated as Eq. (30).

The system is simulated with two state variables ( $x$  and  $y$ ), and the prediction step  $q$  is set to 1. It is essential to compute the Jacobian  $\partial L/\partial \hat{N}$  and  $\partial L/\partial \hat{\Xi}$  to solve Eq. (26). The optimization problem in Eq. (26) utilizes TensorFlow2.7 and the Adam optimizer with a learning rate of 0.001 for calculations. To



**Fig. 8** Comparison results of the augmented SINDy and other methods

promote the sparsity of the identified model, the sequentially thresholded is applied to Eq. (26) for  $N_{loop}$  times.

The initial conjecture of the next iteration was set as the preceding iteration's optimization outcome  $\hat{N}$  in each iteration. Additional parameters for each example, such as the noise level, the initial conditions, the number of iterations  $N_{loop}$  and the sparse threshold  $\lambda$ , are shown in Table 5.

Figure 9 shows the results after applying the proposed method to the dataset with the introduction of both 15% uniform and 15% normal measurement noise. It illustrates that the noise estimation is approximate to the true measurement noise with the form of the single simulation trajectory for the time series, histogram and probability density respectively, as shown in Fig. 9a, b, c, e, respectively. The error metrics for each noise distribution shown in Table 3.

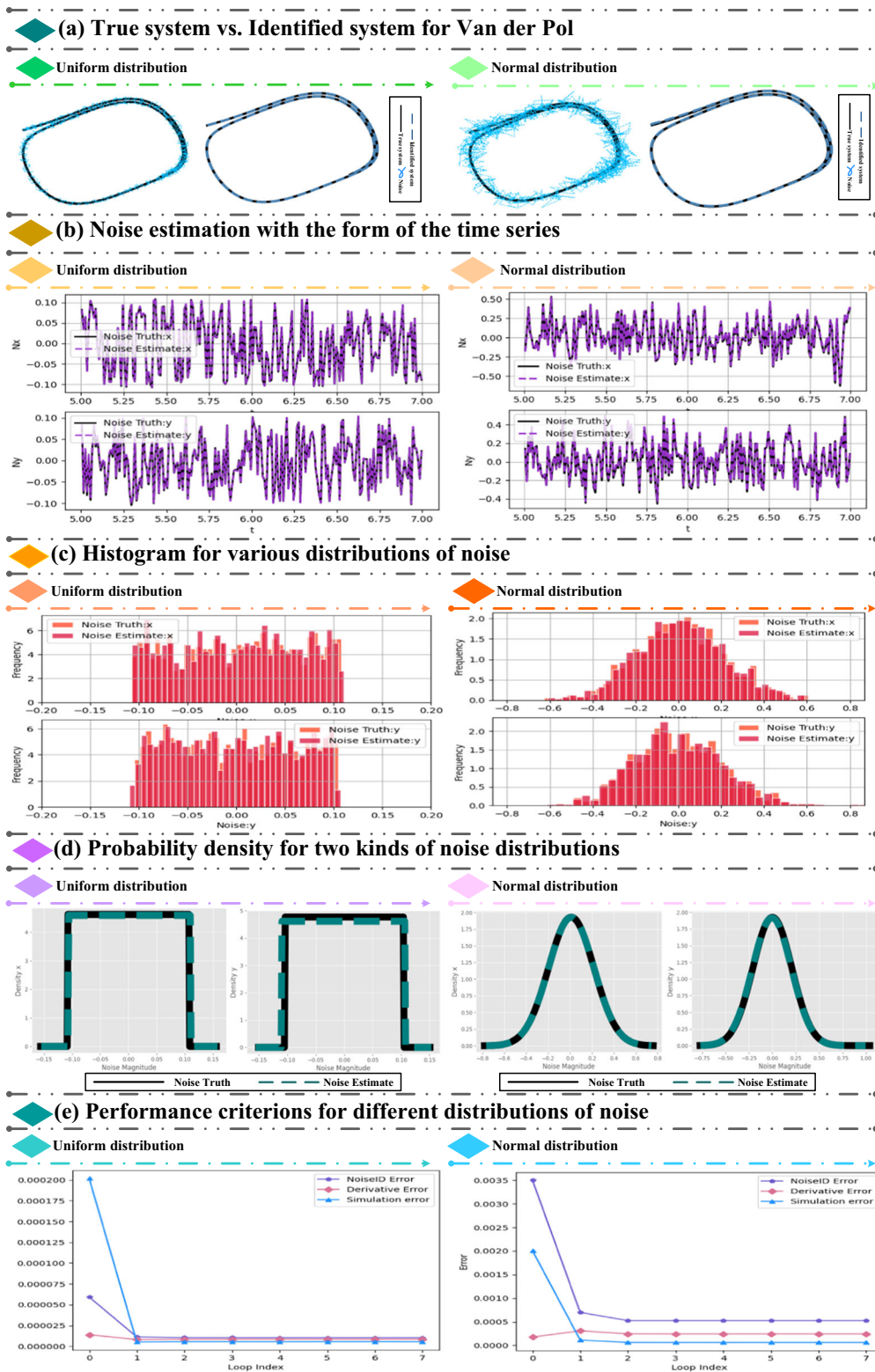
#### 4.4.2 Rössler system

The Rössler system is taken as the final example, whose form is given by Eq. (31).

Data are generated by three input variables ( $x_1, x_2$  and  $x_3$ ), and levels of 30% artificial noise for uniform or normal distribution are added.

Similar to the first case in Sect. 4.4.1, a learning rate of 0.001 of the Adam optimizer and TensorFlow2.7 are executed to learn the models. Auxiliary parameters, including the magnitude of noise, prediction step  $q$ , initial conditions, the amount of iterations  $N_{loop}$  and sparsity threshold  $\lambda$ , are listed in Table 5.

A total of 2500 time steps from  $t = 0$  to  $t = 25$  are generated via high-fidelity simulation and then these data are corrupted with 30% virtual noise in a normal or uniform distribution. The error metrics for the property of the algorithm are outlined in Table 4, and it is shown that the values are approximate to zero.



**Fig. 9** Identification for two kinds of noise distribution and system reconstruction for *van der pol* system

Figure 10 displays the bias in the approximation of the measurement noise for a single trajectory with the time series, the simulation evolving into the attractor and the probability density corrupted by 30% artificial noise. In the left column, the system is contaminated by 30% normal measurement noise. Another column shows the system with 30% uniform noise against the true system.

The learned results, either the simulation trajectory or the probability density of various noise distributions, approach the actual values with minimal error. Additionally, it is evident that the differences for uniform observation noise are smaller than the measurements with 30% normal noise, in such an occasion where the error for 6 iterations declines gradually to approximately 0.

Finally, a range of canonical systems are provided to verify the robustness to significant measurement noise for the algorithm, and their relevant expressions are formulated as Eqs. (30), (31), (34), (35), (36) and (37).

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_1 - 4x_1^3 \end{cases} \quad (34)$$

$$\begin{cases} \dot{x}_1 = 0.5x_1 - 0.025x_1x_2 \\ \dot{x}_2 = -0.5x_2 + 0.05x_1x_2 \end{cases} \quad (35)$$

$$\begin{cases} \dot{x}_1 = 10(x_2 - x_1) \\ \dot{x}_2 = x_1(28 - x_3) - x_2 \\ \dot{x}_3 = x_1x_2 - 8/3x_3 \end{cases} \quad (36)$$

$$\begin{cases} \dot{x}_1 = x_2x_3 \\ \dot{x}_2 = 2(x_1 - x_2) \\ \dot{x}_3 = 5 - x_1x_2 \end{cases} \quad (37)$$

**Table 3** Error metrics for the *Van der Pol* system

Model	Type of noise distribution	Noise percentage (%)	$E_N$	$E_f$	$E_P$
Van der Pol	Normal	15	0.00053	0.00024	6.14E-05
	Uniform	15	1.04E-05	8.49E-06	5.40E-06

**Table 4** Error metrics for the *Rössler* system

Model	Type of noise distribution	Noise percentage (%)	$E_N$	$E_f$	$E_P$
Rössler	Normal	30	0.02272	0.00046	0.01604
	Uniform	30	0.00025	1.63E-06	3.52E-05

The results are boxed in Fig. 11, which shows that the systems are reconstructed successfully with the levels of increasing noise, ranging from 5 to 30%. More broadly, the parameters and the performance indices for these systems are detailed in Tables 5, 6 and 7.

As the prediction step  $q$ , the number of iterations for denoising the data and sparse threshold affect the results of this algorithm, ergodic trials are utilized to undermine the derivative error and to identify the optimum for the abovementioned parameters, with the exception of the overwhelming computational disadvantage.

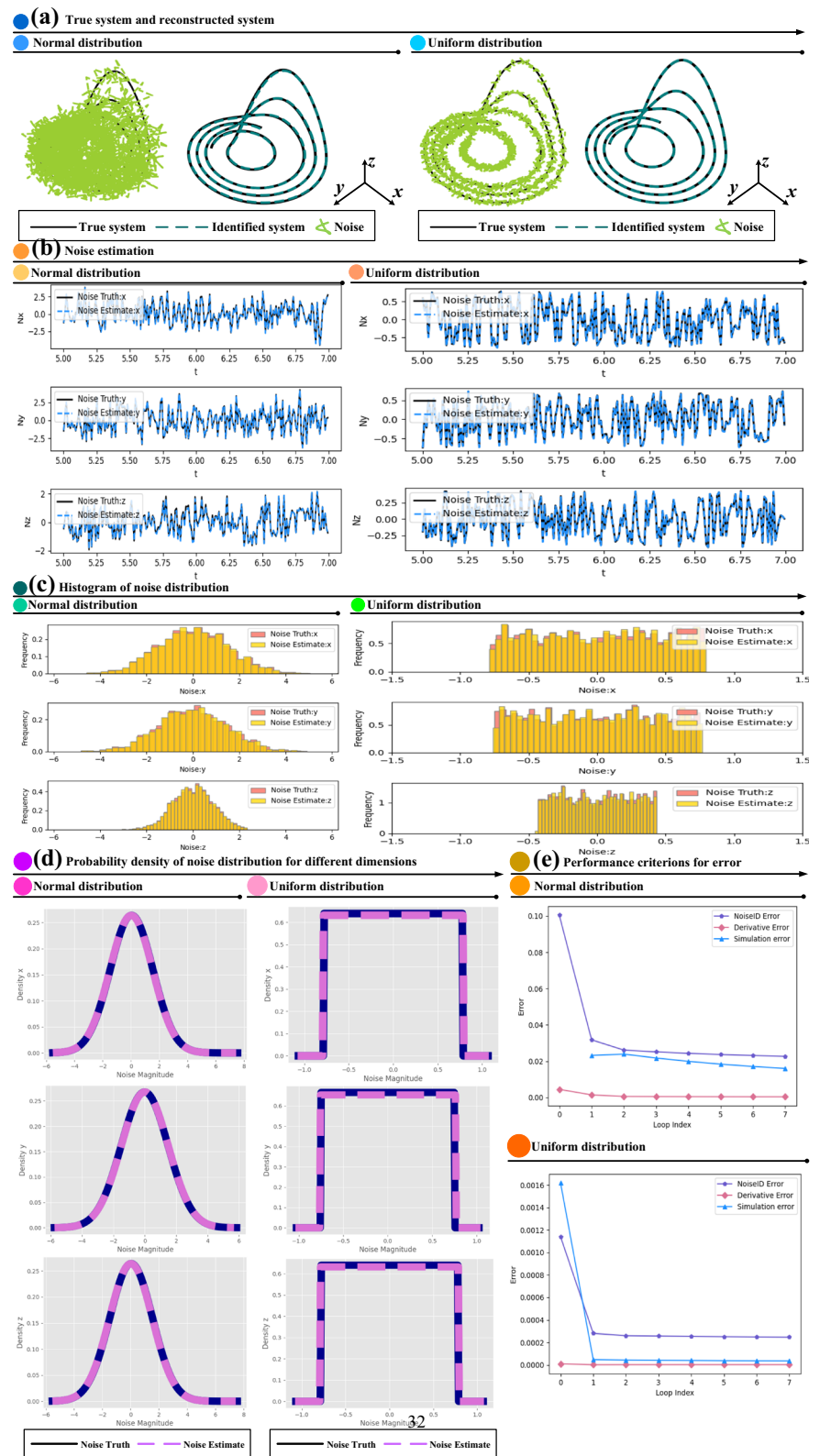
### 5 Discussion and conclusion

The incorporation of mathematical methods is recommended here as follows: (i) identification of nonlinear dynamics via SINDy, (ii) optimal model selection through the AIC, (iii) integration between sparse regression and group sparsity for coefficients of PDEs and (iv) automated identification of noise via data splitting.

From the point of view of experimental results, the proposed method accurately identifies the coefficients for either the ODEs or the PDEs. More broadly, the following limitations may be attracted.

- (1) *Structural analysis.* The clean measurements, the proper candidate function library and the appropriate regularized sparsity threshold are regarded as three preliminary elements for the structure of the SINDy model. Additionally, the

**Fig. 10** The results of identification for both noise distribution with different forms and simulation reconstructions for *Rossler* system





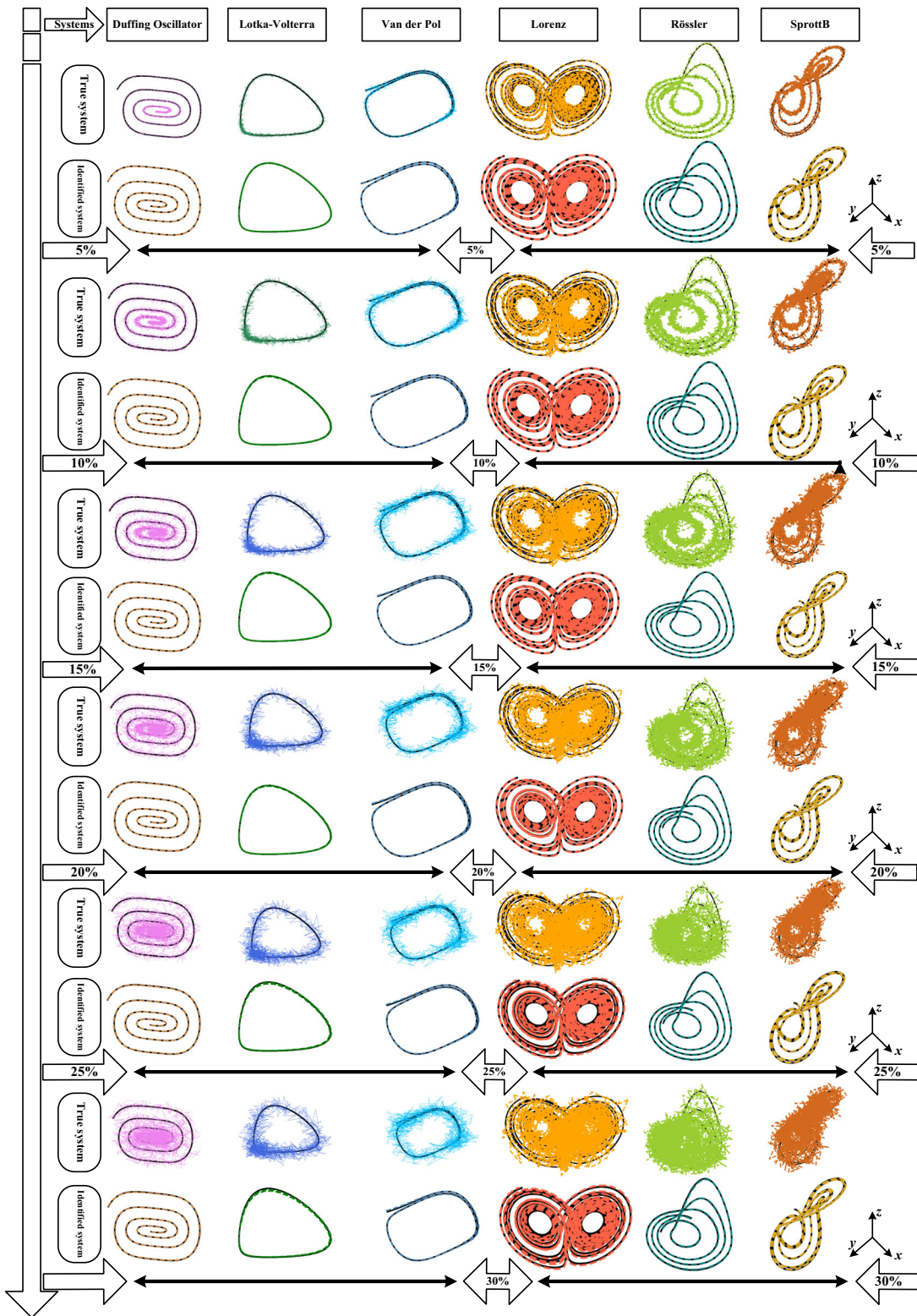


Fig. 11 Corrupted systems and identified systems of a set of levels of artificial noise

**Table 5** A variety of parameters for different test instances

Models	Initial condition	$q$	$\lambda$	$N_{loop}$	Library order	Max Adam optimizer
Duffing	$[-2, 2]$	1	0.05	5	3	5000
Van der Pol	$[-2, 1]$	1	0.15	8	3	5000
Lotka-Volterra	$[1, 2]$	1	0.18	10	3	5000
Lorenz	$[5, 5, 25]$	1	0.20	8	2	15,000
Rössler	$[3, 5, 0]$	1	0.08	8	2	15,000
SprottB	$[1, 1, 1]$	1	0.20	8	2	15,000

**Table 6** Error metrics for 5–15% noise of different test cases

Models	5%			10%			15%		
	$E_N$	$E_f$	$E_P$	$E_N$	$E_f$	$E_P$	$E_N$	$E_f$	$E_P$
Duffing	8.48E-06	4.43E-05	0.0001	3.59E-05	0.00018	0.00042	8.33E-05	0.0004	0.0009
Van der Pol	5.41E-05	1.77E-05	5.07E-06	0.00022	8.76E-05	2.32E-05	0.00053	0.00024	6.14E-05
Lotka-Volterra	0.00073	3.01E-05	4.48E-05	0.00303	0.00016	0.00011	0.03211	0.00018	0.50404
Rössler	0.0005	6.28E-06	0.00045	0.0021	3.10E-05	0.00181	0.00488	7.84E-05	0.00401
Lorenz	0.00484	0.00011	0.48734	0.0193	0.00042	0.46719	0.05444	0.00117	0.44858
SprottB	6.53E-05	1.07E-05	1.53695	0.00026	4.44E-05	1.02298	0.0006	0.00011	1.07388

**Table 7** Error metrics for 20–30% noise of different test cases

Models	20%			25%			30%		
	$E_N$	$E_f$	$E_P$	$E_N$	$E_f$	$E_P$	$E_N$	$E_f$	$E_P$
Duffing	0.00015	0.00067	0.00149	0.00024	0.00101	0.00216	0.00035	0.00138	0.00287
Van der Pol	0.00099	0.00046	0.00012	0.00163	0.00078	0.00019	0.00245	0.00119	0.00029
Lotka-Volterra	0.00765	0.00037	0.00589	0.0183	0.00108	0.00074	0.02883	0.00164	0.00168
Rössler	0.00884	0.00017	0.00722	0.0147	0.00029	0.00029	0.02272	0.00046	0.01604
Lorenz	0.1267	0.00227	0.44827	1.59086	0.01819	0.40318	1.62864	0.01896	0.50842
SprottB	0.00109	0.0002	1.39116	0.00173	0.00032	1.41288	0.00255	0.00048	1.11184

pure measurements play the most important role in the framework; in case noise is introduced, the items in the identified models are bound to be incorrect regardless of the choice of the candidate function library and the sparsity threshold. Furthermore, the sparsity threshold determines the coefficient values of the identified models, and the type of the identified coefficients is defined by potential items in the candidate function library. Due to the actual consideration, the noise is impossible to neglect. Crucially, the indispensable process in sparse regression is parameter tuning.

(2) *Model analysis.* As the structure of the majority of the models in engineering or biological fields is unknown or complex, the class of the items in the candidate function library is inconclusive. Therefore, the solution to the identified model is obscure. More broadly, the threshold selection will undoubtedly be influenced by the polynomial order in the potential function library. As the magnitude of the polynomial order increases, a regularization threshold may be employed to scale down the dimensionality of the function library.



(3) *Robustness analysis.* In fact, there are many kinds of probability distributions of noise, but uniform and normal distributions are showcased in this paper. Moreover, the identified model will not be reconstructed with the introduction of the noise due to the asymmetric structure of the probability distribution, such as the gamma or Rayleigh distribution. The method may be extended to integrate with machine learning models to detect the regime of various probability distributions of noise, particularly those with a nonzero mean.

In conclusion, the proposed method has consequences for data-driven model identification and signal-to-noise separation. The flexibility of the architecture of this approach yields a structure for discovering the underlying dynamics of intricate systems with interpretability. Moreover, it can perhaps be engaged with brain science to exploit epilepsy or Alzheimer’s disease.

**Acknowledgements** The authors would like to thank the valuable comments and suggestions of the anonymous reviewers.

**Author contributions** Conceptualization, XD and YB; methodology, XD; investigation, YL; software, XD; writing—original draft preparation, XD; writing—review and editing, YB; funding acquisition, YB; supervision, YB; validation, MF. All authors have read and agreed to the published version of the manuscript.

**Funding** This study was funded by the NSFC (National Natural Science Foundation of China) project (Grant Numbers: 41861047, 41461078).

**Data availability** Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

**Declarations**

**Conflict of interest** The authors declare that they have no conflict of interest.

**Appendix A**

Table: The summation of the variable

Symbol	Meaning	Symbol	Meaning
<i>State variable</i>		$h$	The decay factor in Sect. 3.4
$\hat{\mathbf{X}}, \mathbf{X}$	The training set in Eq. (2) and Eq. (3)	$\omega_l$	The constrained condition in Eq. (24)
$\hat{\mathbf{X}}, \hat{\mathbf{X}}$	The estimated measurements in Eq. (23)	$\mu(t)$	The parametric dependencies in Eq. (14)
$\mathbf{Y}$	The validation set in Sect. 2	$N_a$	The number of possible monomials in Sect. 4.2
$\mathbf{U}$	The actually observed data in Eq. (20)	$N_c$	The total number of potential models in Sect. 4.2
$\hat{\mathbf{N}}$	The estimated noise in Eq. (26)	$p$	The $p$ th order polynomial library in Sect. 4.2
$\mathbf{X}^{Q_i}$	The $i$ th order of polynomial terms in Eq. (5)	$m$	The dimension of state variable in Sect. 4.2
$\Theta$	The candidate function collection in Eq. (4)	$\lambda$	The threshold in Sect. 4.4
$\mathbf{V}_t$	The measurements for partial differential equations (PDEs) in Eq. (13)		<i>Performance criterion</i>
$\tilde{\Theta}$	The block diagonal matrix $\Theta$ in Eq. (17)	$e_d$	The absolute error of the estimated derivative and system’s vector field in Eq. (23)
$\Xi$	The vector of the sparse coefficients in Eq. (6)	$e_s$	The simulation error in Eq. (25)

Symbol	Meaning	Symbol	Meaning
$\xi_i$	The $i$ th element in the vector of the sparse coefficients in Eq. (7)	$E_N$	The noise identification error in Eq. (27)
$\theta_l$	The $l$ th potential features in the candidate function library in Sect. 2	$E_f$	The vector field error in Eq. (28)
$x_i$	The state variables of distinct dimension in Eq. (30)	$E_p$	The prediction error in Eq. (29)
$v_{\underbrace{xx \dots x}_n}$	The $n$ th order derivative in Eq. (12)	<i>Function</i>	
$t_i$	The time points in Sect. 3.1	$\mathbf{f}(\cdot)$	The function of nonlinear dynamics in Eq. (1)
<i>Parameter</i>		$\mathbf{F}(\cdot)$	The flow map of the dynamic systems in Eq. (21)
$\Gamma$	A collection of groups in Eq. (18)	RSS	The residual sum of squares in Sect. 3.2
$\hat{\mu}$	The estimated parameter in Eq. (10)	$L(\cdot)$	The loss function in Eq. (26)
$k$	The degree free parameters in Eq. (10)	$\Delta_i$	The relative AICc in Sect. 4.2
$l$	The dimension of measurements in Eq. (11)	$AIC_{\min}$	The minimum AIC value in Sect. 3.2
$\varepsilon$	The noise magnitude in Sect. 3.2	$R(\cdot)$	The regularization function in Eq. (9)
$q$	The backward or forward time steps in Eq. (22)	$N(\cdot)$	The nonlinear function for the dynamics in Eq. (12)

## References

- Schmidt, M., Lipson, H.: Distilling free-form natural laws from experimental data. *Science* **324**(5923), 81–85 (2009)
- Brunton, S.L., Kutz, J.N.: *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press (2019)
- Budišić, M., Mohr, R., Mezić, I.: Applied koopmanism. *Chaos* **22**(4), 047510 (2012)
- Mezić, I.: Analysis of fluid flows via spectral properties of the Koopman operator. *Annu. Rev. Fluid Mech.* **45**, 357–378 (2013)
- Klus, S., Nüske, F., Koltai, P., et al.: Data-driven model reduction and transfer operator approximation. *J. Nonlinear Sci.* **28**(3), 985–1010 (2018)
- Schmid, P.J.: Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **656**, 5–28 (2010)
- Raissi, M., Yazdani, A., Karniadakis, G.E.: Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. *Science* **367**(6481), 1026–1030 (2020)
- Mardt, A., Pasquali, L., Wu, H., et al.: VAMPnets for deep learning of molecular kinetics. *Nat. Commun.* **9**(1), 1–11 (2018)
- Dylewsky, D., Barajas-Solano, D., Ma, T., et al.: Stochastically forced ensemble dynamic mode decomposition for forecasting and analysis of near-periodic systems (2020). arXiv preprint [arXiv:2010.04248](https://arxiv.org/abs/2010.04248)
- Kevrekidis, I.G., Gear, C.W., Hyman, J.M., et al.: Equation-free, coarse-grained multiscale computation: enabling microscopic simulators to perform system-level analysis. *Commun. Math. Sci.* **1**(4), 715–762 (2003)
- Giannakis, D., Majda, A.J.: Nonlinear Laplacian spectral analysis for time series with intermittency and low-frequency variability. *Proc. Natl. Acad. Sci.* **109**(7), 2222–2227 (2012)
- Baddoo, P.J., Herrmann, B., McKeon, B.J., et al.: Kernel learning for robust dynamic mode decomposition: linear and nonlinear disambiguation optimization (LANDO) (2021). arXiv preprint [arXiv:2106.01510](https://arxiv.org/abs/2106.01510)
- Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B* **58**(1), 267–288 (1996)
- Zhang, L., Schaeffer, H.: On the convergence of the SINDy algorithm. *Multiscale Model. Simul.* **17**(3), 948–972 (2019)
- Boninsegna, L., Nüske, F., Clementi, C.: Sparse learning of stochastic dynamical equations. *J. Chem. Phys.* **148**(24), 241723 (2018)
- Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. *SIAM Rev.* **43**(1), 129–159 (2001)
- Mallat, S.G., Zhang, Z.: Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* **41**(12), 3397–3415 (1993)
- Kullback, S., Leibler, R.A.: On information and sufficiency. *Ann. Math. Stat.* **22**(1), 79–86 (1951)
- Akaike, H.: Information theory and an extension of the maximum likelihood principle. In: *Selected Papers of Hirotugu Akaike*, pp. 199–213. Springer, New York (1998)
- Akaike, H.: A new look at the statistical model identification. *IEEE Trans. Autom. Control* **19**(6), 716–723 (1974)
- Schwarz, G.: Estimating the dimension of a model. *Ann. Stat.* **6**, 461–464 (1978)
- Bishop, C.M., Nasrabadi, N.M.: *Pattern recognition and machine learning*, vol. 4, no. 4. Springer, New York (2006)
- Van Der Linde, A.: DIC in variable selection. *Stat. Neerl.* **59**(1), 45–56 (2005)

24. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci.* **113**(15), 3932–3937 (2016)
25. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Sparse identification of nonlinear dynamics with control (SINDYc). *IFAC-PapersOnLine* **49**(18), 710–715 (2016)
26. Mangan, N.M., Brunton, S.L., Proctor, J.L., et al.: Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Trans. Mol. Biol. Multi-Scale Commun.* **2**(1), 52–63 (2016)
27. Qu, Q., Sun, J., Wright, J.: Finding a sparse vector in a subspace: linear sparsity using alternating directions. *IEEE Trans. Inf. Theory* **62**(10), 5855–5880 (2016)
28. Lorenz, E.N.: Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**(2), 130–141 (1963)
29. Quade, M., Abel, M., Nathan Kutz, J., et al.: Sparse identification of nonlinear dynamics for rapid model recovery. *Chaos* **28**(6), 063116 (2018)
30. Bramburger, J.J., Kutz, J.N., Brunton, S.L.: Data-driven stabilization of periodic orbits. *IEEE Access* **9**, 43504–43521 (2021)
31. Champion, K., Lusch, B., Kutz, J.N., et al.: Data-driven discovery of coordinates and governing equations. *Proc. Natl. Acad. Sci.* **116**(45), 22445–22451 (2019)
32. Kaptanoglu, A.A., Morgan, K.D., Hansen, C.J., et al.: Physics-constrained, low-dimensional models for magnetohydrodynamics: first-principles and data-driven approaches. *Phys. Rev. E* **104**(1), 015206 (2021)
33. Wei, B.: Sparse dynamical system identification with simultaneous structural parameters and initial condition estimation (2022). arXiv preprint [arXiv:2204.10472.v](https://arxiv.org/abs/2204.10472)
34. Zheng, P., Askham, T., Brunton, S.L., et al.: A unified framework for sparse relaxed regularized regression: Sr3. *IEEE Access* **7**, 1404–1423 (2018)
35. Champion, K., Zheng, P., Aravkin, A.Y., et al.: A unified sparse optimization framework to learn parsimonious physics-informed models from data. *IEEE Access* **8**, 169259–169271 (2020)
36. Fasel, U., Kutz, J.N., Brunton, B.W., et al.: Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control (2021). arXiv preprint [arXiv:2111.10992](https://arxiv.org/abs/2111.10992)
37. Hirsh, S.M., Barajas-Solano, D.A., Kutz, J.N.: Sparsifying priors for Bayesian uncertainty quantification in model discovery. *R. Soc. Open Sci.* **9**(2), 211823 (2022)
38. Ram, P.R.M., Römer, U., Semaan, R.: Bayesian dynamical system identification with unified sparsity priors and model uncertainty (2021). arXiv preprint [arXiv:2103.05090](https://arxiv.org/abs/2103.05090)
39. Mangan, N.M., Askham, T., Brunton, S.L., et al.: Model selection for hybrid dynamical systems via sparse regression. *Proc. R. Soc. A* **475**(2223), 20180534 (2019)
40. Jiang, Y.X., Xiong, X., Zhang, S., et al.: Modeling and prediction of the transmission dynamics of COVID-19 based on the SINDy-LM method. *Nonlinear Dyn.* **105**(3), 2775–2794 (2021)
41. Sorokina, M., Sygletos, S., Turitsyn, S.: Sparse identification for nonlinear optical communication systems: SINO method. *Opt. Express* **24**(26), 30433–30443 (2016)
42. Fukami, K., Murata, T., Zhang, K., et al.: Sparse identification of nonlinear dynamics with low-dimensionalized flow representations. *J. Fluid Mech.* (2021). <https://doi.org/10.1017/jfm.2021.697>
43. Loiseau, J.C.: Data-driven modeling of the chaotic thermal convection in an annular thermosyphon. *Theor. Comput. Fluid Dyn.* **34**(4), 339–365 (2020)
44. Wu, Y.: Error Processing of sparse identification of nonlinear dynamical systems via  $L_\infty$  approximation (2021). arXiv preprint [arXiv:2107.06142](https://arxiv.org/abs/2107.06142)
45. de Silva, B., Champion, K., Quade, M., et al.: PySINDy: A Python package for the sparse identification of nonlinear dynamical systems from data. *JOSS* **5**, 2104 (2020)
46. Kaptanoglu, A.A., de Silva, B.M., Fasel, U., et al.: PySINDy: a comprehensive Python package for robust sparse system identification (2021). arXiv preprint [arXiv:2111.08481](https://arxiv.org/abs/2111.08481)
47. Kaiser, E., Kutz, J.N., Brunton, S.L.: Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. A* **474**(2219), 20180335 (2018)
48. Hoffmann, M., Fröhner, C., Noé, F.: Reactive SINDy: discovering governing reactions from concentration data. *J. Chem. Phys.* **150**(2), 025101 (2019)
49. Chu, H.K., Hayashibe, M.: Discovering interpretable dynamics by sparsity promotion on energy and the Lagrangian. *IEEE Robot. Autom. Lett.* **5**(2), 2154–2160 (2020)
50. Dai, M., Gao, T., Lu, Y., et al.: Detecting the maximum likelihood transition path from data of stochastic dynamical systems. *Chaos* **30**(11), 113124 (2020)
51. Brunton, S.L., Brunton, B.W., Proctor, J.L., et al.: Chaos as an intermittently forced linear system. *Nat. Commun.* **8**(1), 1–9 (2017)
52. Loiseau, J.C., Noack, B.R., Brunton, S.L.: Sparse reduced-order modelling: sensor-based dynamics to full-state estimation. *J. Fluid Mech.* **844**, 459–490 (2018)
53. Kaheman, K., Brunton, S.L., Kutz, J.N.: Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data. *Mach. Learn.* **3**(1), 015031 (2022)
54. Dam, M., Brøns, M., Juul Rasmussen, J., et al.: Sparse identification of a predator–prey system from simulation data of a convection model. *Phys. Plasmas* **24**(2), 022310 (2017)
55. Messenger, D.A., Bortz, D.M.: Weak SINDy for partial differential equations. *J. Comput. Phys.* **443**, 110525 (2021)
56. Cortiella, A., Park, K.C., Doostan, A.: Sparse identification of nonlinear dynamical systems via reweighted  $\ell_1$ -regularized least squares. *Comput. Methods Appl. Mech. Eng.* **376**, 113620 (2021)
57. Bhadriraju, B., Narasingam, A., Kwon, J.S.I.: Machine learning-based adaptive model identification of systems: application to a chemical process. *Chem. Eng. Res. Des.* **152**, 372–383 (2019)
58. Jadhav, Y., Barati Farimani, A.: Dominant motion identification of multi-particle system using deep learning from video. *Neural Comput. Appl.* (2022). <https://doi.org/10.1007/s00521-022-07421-z>
59. Takens, F.: Detecting strange attractors in turbulence. In: *Dynamical Systems and Turbulence*, Warwick 1980, pp:366–381. Springer, Berlin (1981)
60. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D* **60**(1–4), 259–268 (1992)

61. Burnham, K.P., Anderson, D.R.: Multimodel inference: understanding AIC and BIC in model selection. *Sociol. Methods Res.* **33**(2), 261–304 (2004)
62. Rudy, S.H., Brunton, S.L., Proctor, J.L., et al.: Data-driven discovery of partial differential equations. *Sci. Adv.* **3**(4), e1602614 (2017)
63. Schaeffer, H., Tran, G., Ward, R.: Learning dynamical systems and bifurcation via group sparsity (2017). arXiv preprint [arXiv:1709.01558](https://arxiv.org/abs/1709.01558)
64. Daniels, B.C., Nemenman, I.: Automated adaptive inference of phenomenological dynamical models. *Nat. Commun.* **6**(1), 1–8 (2015)
65. Rudy, S.H., Kutz, J.N., Brunton, S.L.: Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. *J. Comput. Phys.* **396**, 483–506 (2019)
66. Van Der Pol, B.: Vii forced oscillations in a circuit with non-linear resistance. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **3**(13), 65–80 (1927)
67. Rössler, O.E.: An equation for continuous chaos. *Phys. Lett. A* **57**(5), 397–398 (1976)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.