Supporting Information for *Adaptive backward stepwise selection of fast sparse identification of nonlinear dynamics*

Feng JIANG, Lin DU, Qing XUE, Zichen DENG, Celso GREBOGI

August 28, 2025

Contents

1 Algorithm and implementation

1.1 Adaptive Backward Stepwise Selection SINDy (ABSS-SINDy)

The more detailed pseudocode implementation of our key optimization procedure is presented in Algorithm 1, which is based on MATLAB.

Algorithm 1 *The key optimization procedure in our ABSS-SINDy.*

```
procedure [Ξ, η, ω] = SparseDynamics_ABSS(Θ, dX)
    % Input:
    %    Θ ∈ R^{N×p}: the candidate basis matrix.
    %    dX ∈ R^{N×n}: the derivative matrix.
    % Output:
    %    Ξ ∈ R^{p×n}: the sparse solution matrix.
    %    η ∈ R^n: the threshold value vector.
    %    ω ∈ R^n: the iteration vector.
    objective = @(A, x, b) 1/2*norm(A*x-b).^2;          ▷ Objective function.
    ρ ← 1e − 6;                                          ▷ The default ridge regression coefficient.
    A ← Θ;
    for i ← 1, n do                                      ▷ Splitting into n subproblems firstly.
        b ← dX(:, i);
        x ← ridge_regression(A, b, ρ);                   ▷ The ridge regression estimation as the initial solution.
        R ← [ ];                                         ▷ Record the redundant coefficients.
        ind ← logical(1 : p);
        for k ← 1, p − 1 do
            x_old ← x;
            % Core of our backward stepwise selection (five line code marked in red).
            x_min = min(abs(x(ind)));                    ▷ Obtain the smallest coefficient.
            threshold ← x_min + ϵ;                       ▷ Set temporary threshold value.
            ind ← abs(x) > threshold;                    ▷ Choose items with big Coefficients.
            x(∼ ind) ← 0;                                ▷ Small coefficients are set to 0.
            x(ind) ← ridge_regression(A(:, ind), b, ρ);  ▷ Ridge regression estimation again.
            % Calculate the modified relative error.
            obj_old ← objective(A, x_old, b);
            obj_new ← objective(A, x, b);
            MRE ← calModifiedRelativeError(obj_old, obj_new);
            R ← [R, threshold];
            % Stopping criterion 1: abrupt change.
            if MRE > 0.01 then                           ▷ The value here is set empirically.
                ω(i) ← k − 1;
                η(i) ← max(R(1 : k − 1));
                x ← x_old;
                Break;
            end if
            % Stopping criterion 2: flat change to the base model.
            if sum(ind) == 1 then
                ω(i) ← k;
                η(i) ← max(R(1 : k));
            end if
        end for
        Ξ(:, i) ← x;
    end for
end procedure
```

1.2 Dictionary learning-based Fast SINDy

1.2.1 Basic motivation and effectiveness

In the introduction of the manuscript, we briefly pointed out the motivation for our study. Here, we can elaborate on the relevant content.

The curse of dimensionality presents a significant technical challenge when simulating high-dimensional systems. To illustrate this phenomenon, this study utilizes PySINDy, a well-known Python package in the field of sparse identification that has already integrated various methodologies. The curse of dimensionality poses a significant challenge, especially as the system dimension grows and simulation time increases. For instance, using the latest package with a purely polynomial basis of maximum order 3, the simulation times for the Lorenz 96 model are: 105.05 seconds (4D), 184.03 seconds (5D), 267.23 seconds (6D), and 339.49 seconds (7D). This time-consuming process is primarily due to the symbolic regression required for each integration, which limits the application of SINDy in high-dimensional problems. Recently, Egan et al. made a comparison that demonstrates the computational efficiency by using two low-dimensional models: the two-dimensional linear model and the three-dimensional Lorenz 63 model. The results indicate that the computational time varies from 100 seconds to 10,000 seconds, further verifying that the current computational efficiency remains an urgent problem to be solved. To address this issue, we propose the idea of dictionary learning. In the SINDy framework, the basis library is constructed in advance, and sparse optimization is applied to this library. Since all the features of the basis library are known, it can be treated as a form of dictionary learning. By decoding the features of the basis library into text, we can construct the entire expression of the identified function, eliminating the need for symbolic regression during each integration process. This approach underpins our technical solution to the curse of dimensionality. By applying this modified PySINDy package incorporating dictionary learning, the simulation times for the Lorenz 96 model, using a purely polynomial basis of maximum order 3, are reduced to 0.47 seconds (4D), 0.55 seconds (5D), 0.58 seconds (6D), and 0.54 seconds (7D). Remarkably, the simulation time for a 20-dimensional Lorenz 96 model is only 1.00 seconds, demonstrating exceptional performance in handling high-dimensional problems.

1.2.2 Technical implementation

There are two major computational platforms in the field of sparse identification of nonlinear dynamics: MATLAB and Python. Therefore, we focus on implementing methods to overcome the curse of dimensionality on these platforms. Enhancing simulation speed is crucial for high-dimensional systems, and for simplicity, we consider only the purely polynomial basis library. SINDy and related methods depend on a pre-defined basis library, which can be treated as a known dictionary and decoded into a textual format easily computable on the respective platforms. After performing sparse optimization based on this basis library, we can decode the identified coefficients, combine the decoded dictionary, and transform the results into a functional form compatible with MATLAB or Python.

For example, the Lorenz 96 model with the dimension of 4 and the polynomial basis with the maximum order of 2 (see Fig. 1). To unify the expression in different platforms, the state variable is expressed as $x_i\,(i = 1, 2, 3, 4)$. Firstly, we consider the MATLAB platform. The polynomial features with the order of 1 are expressed as $x_i\,(i = 1, 2, 3, 4)$, and the polynomial features with the order of 2 are expressed as $x_ix_j\,(i = 1, 2, 3, 4; j = 1, 2, 3, 4; i \leq j)$. Those are previously defined features. Because MATLAB uses parentheses to index vectors, we can decode the features into "$X(i)$" $(i = 1, 2, 3, 4)$ and "$X(i) * X(j)$" $(i = 1, 2, 3, 4; j = 1, 2, 3, 4; i \leq j)$. Considering the coefficients, the corresponding features can be transformed into "$+|c| * X(i)$", "$-|c| * X(i)$", "$+|c| * X(i) * X(j)$", or "$-|c| * X(i) * X(j)$". Then, we can obtain the textual equations of the identified system and, finally, the whole textual function in MATLAB.

There are slight differences for the Python platform. Because the array index in Python starts from 0 and Python uses square brackets to index the list, we can define the state variable as $x_i\,(i = 0, 1, 2, 3)$. The polynomial features with the order of 1 are defined as $x_i\,(i = 0, 1, 2, 3)$ and the polynomial features with the order of 2 are defined as $x_ix_j\,(i = 0, 1, 2, 3; j = 0, 1, 2, 3; i \leq j)$. Similarly, considering the coefficients, the final corresponding features can be transformed into "$+|c| * X[i]$", "$-|c| * X[i]$", "$+|c| * X[i] * X[j]$", or "$-|c| * X[i] * X[j]$".

Why do we need this procedure? When analyzing the limitation of the F-test, we realized the necessity of doing high-dimensional numerical experiments, which is an urgent desire to push us to solve the curse of dimensionality, reflected in the long simulation time of high-dimensional systems. The solution is motivated by the idea of dictionary learning, and it remarkably reduces the simulation time, making the SINDy run very fast. Consequently, we term this method Dictionary Learning-based Fast SINDy (DL-FSINDy). This method can be treated as a single module for accelerating SINDy. Although the solution is necessary and effective in high-dimensional systems, it also promotes the simulation effectiveness of low-dimensional systems. Generally speaking, it is a general method for reconstructing any dimensional system. Additional technical details and examples are available in the source codes. Implementing DL-FSINDy on other computational platforms, such as R and Julia, involves the same procedure outlined above, with the primary consideration being the definition of functions on those platforms.

The demonstration of the curse of dimensionality is shown in both Python and MATLAB. The implementation of the Dictionary Learning-based Fast SINDy (FSINDy) module is performed on both computational platforms. A direct graphical illustration shows why symbolic regression-based simulation is slow and why FSINDy-based simulation is fast. Here, partial programs from the original SINDy and our developed algorithms are presented. It intuitively explains that the simulation method based on symbolic regression needs to calculate symbolic expressions at each integration step. However, our improved method only requires pre-encoding and decoding processes to share a unified expression at each integration step. This process will significantly reduce the time required for the simulation, especially in the reconstruction of high-dimensional systems. The specific experimental results can be viewed in the main text of the manuscript and by running the source code we provided for direct experience.

1.3 Code availability

All source code and data are available on GitHub at https://github.com/smallFF/ABSS-FSINDy. Due to the variation in the running time of the code across different computers, the numerical values regarding time efficiency may differ. However, the main conclusion remains consistent with this article.

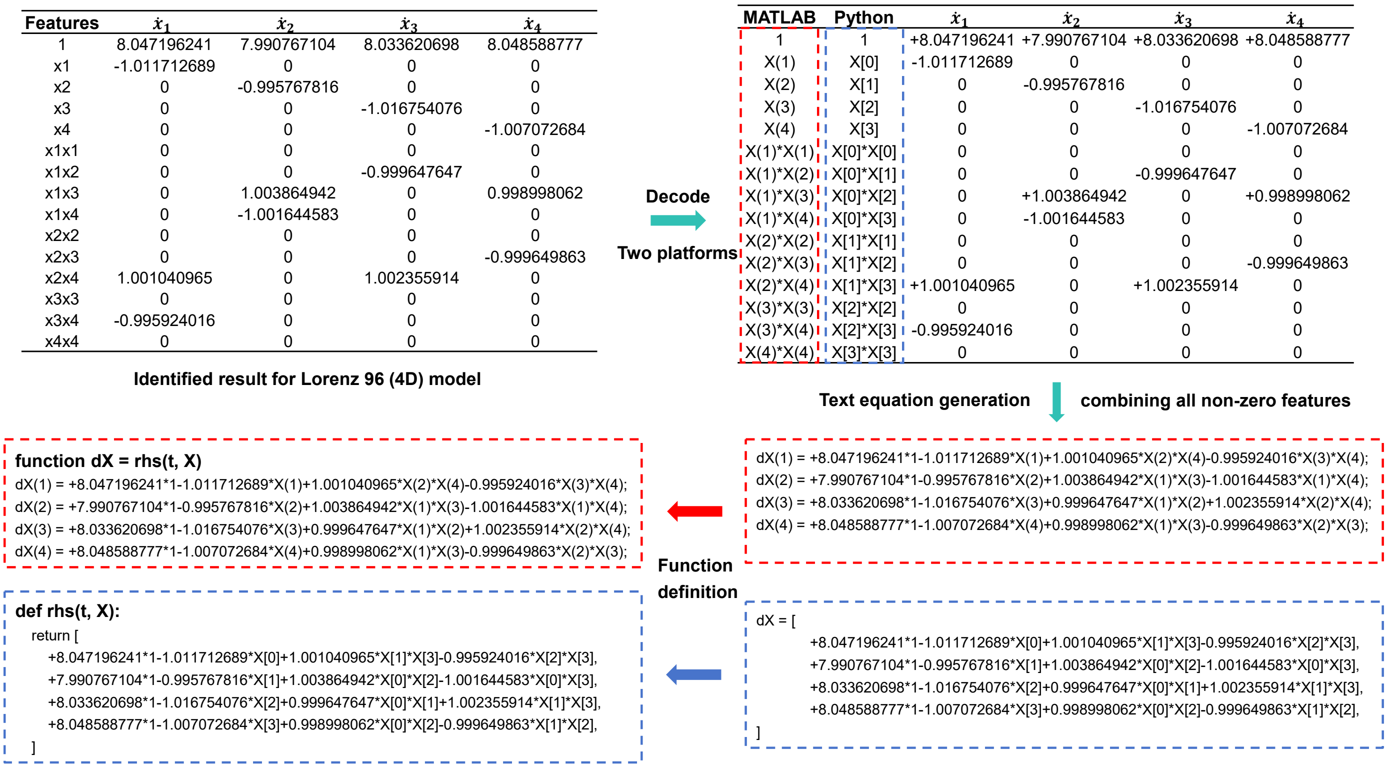


Figure 1: An example illustrating how the concept of dictionary learning can be used to accelerate simulations on two different computational platforms—MATLAB and Python.

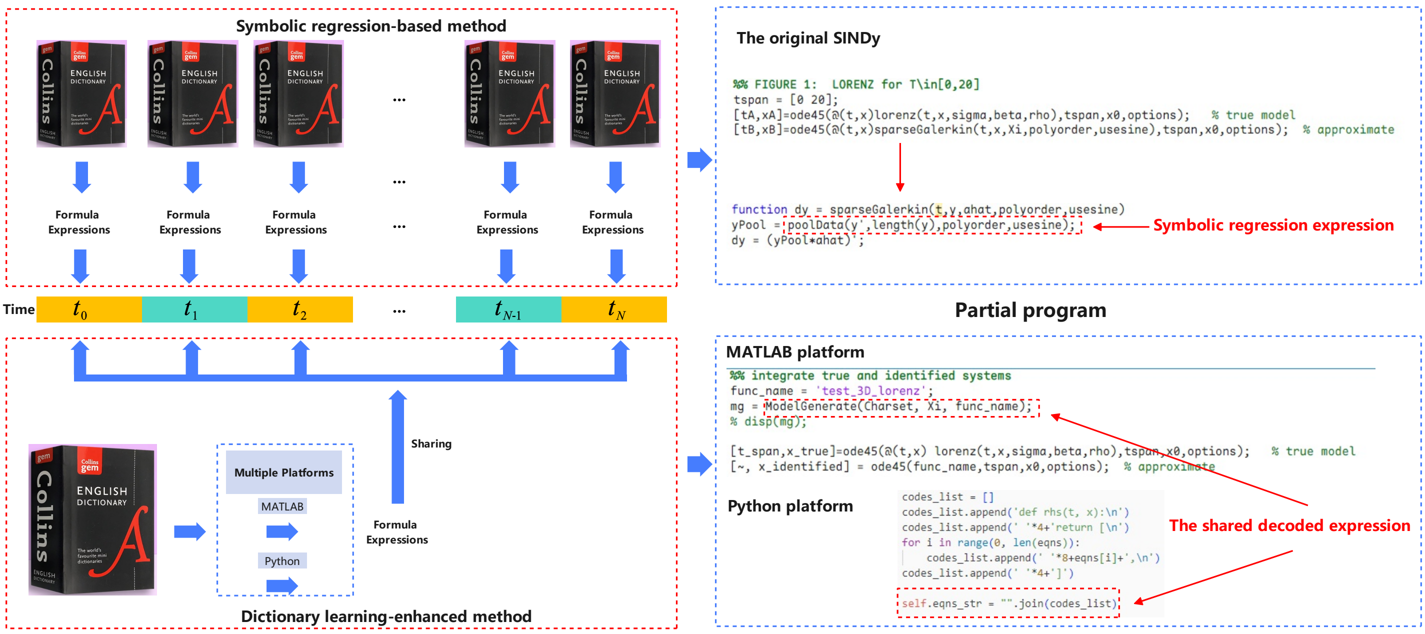


Figure 2: The graphical comparison of the symbolic regression-based simulation and the FSINDy-based simulation.

Type	System	Equation	Parameters	$k_{\max}$	λ
	Linear (2D)	$\dot{x} = -\lambda x + \mu y$	$\lambda = 0.1$	8	0.003175
		$\dot{y} = -\mu x - \lambda y$	$\mu = 2$	8	0.002358
	Cubic (2D)	$\dot{x} = -\lambda x^3 + \mu y^3$	$\lambda = 0.1$	19	0.004516
		$\dot{y} = -\mu x^3 - \lambda y^3$	$\mu = 2$	19	0.006696
	van der Pol (2D)	$\dot{x} = y$	$\mu = 2$	9	0.001355
		$\dot{y} = -x - \mu(x^2 - 1)y$		7	0.000574
	Linear (3D)	$\dot{x} = -\lambda x + \mu y$	$\lambda = 0.1$	8	0.001916
LD		$\dot{y} = -\mu x - \lambda y$	$\mu = 2$	8	0.004007
		$\dot{z} = -\rho z$	$\rho = 0.3$	9	0.000998
	Rössler (3D)	$\dot{x} = -(y + z)$	$\alpha = 0.2$	18	0.004441
		$\dot{y} = x + \alpha y$	$\beta = 5.7$	18	0.016392
		$\dot{z} = \alpha + z(x - \beta)$		17	0.005724
	Lorenz 63 (3D)	$\dot{x} = -\sigma(y - x)$	$\sigma = 10$	18	0.001404
		$\dot{y} = x(\rho - z) - y$	$\beta = 8/3$	17	0.002267
		$\dot{z} = xy - \beta z$	$\rho = 28$	18	0.001131
	Lotka-Volterra (4D)	$\dot{x} = x(r_1 - x - ay - bz)$	$r_1 = r_2 = r_3 = r_4 = 1$	11	0.014183
		$\dot{y} = y(r_2 - y - z - w)$	$a = 2$	11	0.002159
		$\dot{z} = z(r_3 - cx - z)$	$b = 1.3$	12	0.005392
		$\dot{w} = w(r_4 - x - w)$	$c = 2$	12	0.003555
	Rössler Hyperchaotic (4D)	$\dot{x} = -y - z$	$a = 0.25$	13	0.000016
		$\dot{y} = x + ay + w$	$b = 3$	12	0.000088
		$\dot{z} = b + xz$	$c = 0.05$	13	0.000004
		$\dot{w} = cw + dz$	$d = 0.5$	13	0.000080
LD	Coupled Lorenz system (6D)	$\dot{x} = -x + y + ku$		25	0.000058
		$\dot{y} = -xz$		27	0.000060
		$\dot{z} = -R + xy$	$k = 0.5$	26	0.000066
		$\dot{u} = -u + v + kx$	$R = 1$	25	0.000057
		$\dot{v} = -uw$		27	0.000054
		$\dot{w} = uv - R$		26	0.000072
	Lorenz 96 (4D to 23D)	$\dot{x_i} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F$	$F = 8$	$k_{\max}(n)$	$\lambda(n)$
		for $i = 1, 2, \cdots, n$	$n = 4, 5, \ldots, 23$		
		where $x_{i-n} = x_{i+n} = x_i$			

Table 1: The results identified using ABSS-SINDy, where $k_{\max}$ represents the maximum number of iterations (one hyperparameter), and λ is the threshold value (another hyperparameter). The hyperparameters depend on the system dimension n for the high-dimensional system.

2 Supplementary results

This Section presents all the omitted results for the examples in the manuscript (see Table 1). The linear (3D) model is typical to demonstrate the two stopping criteria, so other examples are shown in this Section. Both of them share the same form like the form of the linear (3D) model in the manuscript, including the changes in objective value and modified relative error with the backward stepwise selection, the absolute value of coefficients on the same axis, the quantitative comparison between the true and identified systems, and the comparison of the coefficients between the true and identified systems.

Here, we use the two error metrics—the *root mean square error* (RMSE) and the *finite prediction time* (FPT)—to measure the error between the true system and the identified system for non-chaotic and chaotic cases. Given the observation data set $\{s_i \mid i = 1, 2, \cdots, N\}$ and the prediction data set $\{\widehat{s}_i \mid i = 1, 2, \cdots, N\}$, the RMSE is defined as follows:

$$\mathrm{RMSE}(\widehat{s}, s) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\widehat{s}_i - s_i)^2}, \tag{1}$$

and the FPT is defined as follows:

$$\mathrm{FPT}(\boldsymbol{s}, \widehat{\boldsymbol{s}}, \epsilon) = \arg\min_t \left\{t : |\boldsymbol{s}(t) - \widehat{\boldsymbol{s}}(t)| >= \epsilon, \forall t \in [0, T]\right\}, \tag{2}$$

where ϵ represents the significant level of separation between the two systems in a chaotic case.

2.1 2D: The linear equation (positive example)

The linear equation is as follows:

$$\begin{cases} \dot{x_1} = -\lambda x_1 + \mu x_2, \\ \dot{x_2} = -\mu x_1 - \lambda x_2. \end{cases} \tag{3}$$

In this numerical experiment, we set parameters $\lambda = 0.1$ and $\mu = 2$, the simulation time interval $T = [0, 25]$, the step size $dt = 0.01$, the initial condition $X(0) = [2, 0]^\top$, the noise standard deviation $\epsilon = 0.05$ and the order of the polynomial basis library $d = 3$.

The results are summarized in three figures and one table, which collectively demonstrate the performance of ABSS-SINDy in identifying the system dynamics. Figure 3 shows the changes in objective value and modified relative error with the backward stepwise selection. Figure 4 shows the absolute value of coefficients on the same axis. Figure 5 shows the quantitative comparison between the true and identified systems. Table 2 compares the coefficients between the true and the identified systems.

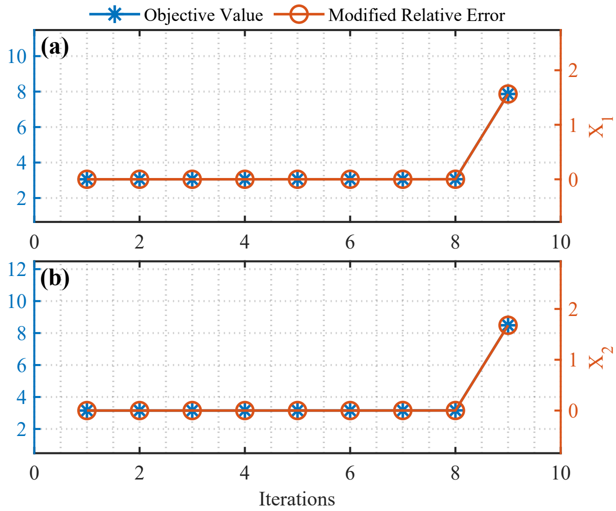
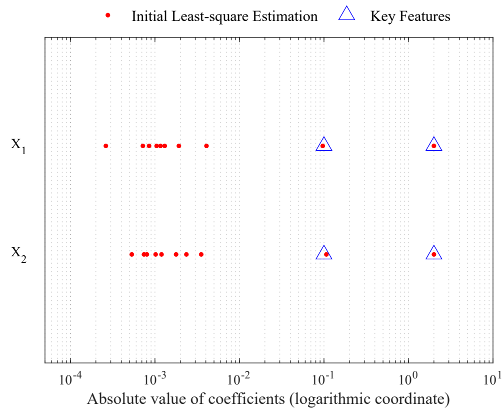
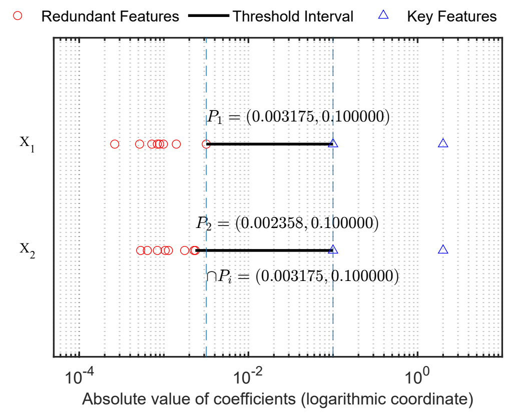


Figure 3: The process of the backward stepwise selection (positive example). The left y-axis represents the object value, and the right y-axis represents the MRE. Here, we construct the polynomial basis with the maximum order $d = 3$, and due to the state variables $n = 2$, the number of basis functions $p = \binom{n+d}{d} = 10$. The number of key features are $[n_{kf}]_1 = 2$ and $[n_{kf}]_2 = 2$. The hyperparameter $k_{\max}$ defined here will be accurately equal to the number of total features minus the number of key features: (a) $[k_{\max}]_1 = 8 = p - [n_{kf}]_1$; and (b) $[k_{\max}]_2 = 8 = p - [n_{kf}]_2$.



(a) Initial estimation based on least-square



(b) Final estimation based on backward stepwise selection

Figure 4: The absolute value of coefficients on the same axis (positive example). a) The red dots represent the initial estimated coefficients based on least squares. The blue triangles represent the key features of the system. b) The red circles represent the redundant features because the features will be deleted during the process of the backward stepwise selection. The blue triangles represent the key features of the system.

Table 2: The Identified Linear (2D) System

Structures	**Coefficients (True)**		**Coefficients (Identified)**	
	$\dot{\mathbf{x}}_1$	$\dot{\mathbf{x}}_2$	$\dot{\hat{\mathbf{x}}}_1$	$\dot{\hat{\mathbf{x}}}_2$
x_1	-0.1	-2	-0.098255	-1.997303
x_2	2	-0.1	1.997647	-0.103797

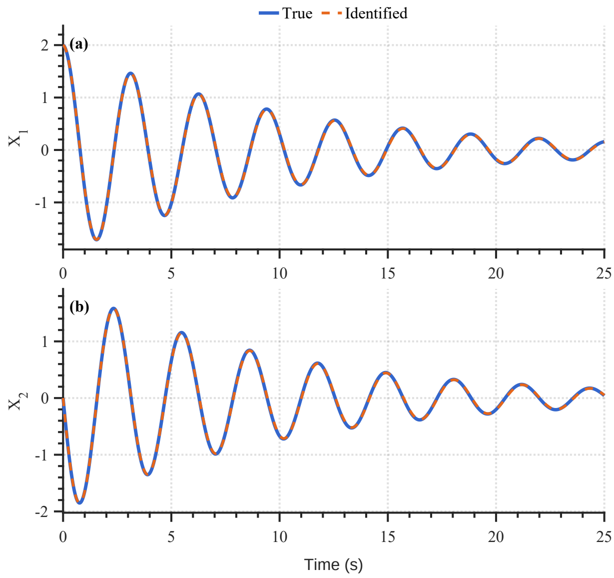


Figure 5: The quantitative comparison, measured by RMSE, between the true and the identified systems: (a) RMSE= 0.011968; and (b) RMSE = 0.011323.

2.2 2D: The cubic equation (positive example)

The cubic equation is as follows:

$$\begin{cases} \dot{x_1} = -\lambda x_1^3 + \mu x_2^3, \\ \dot{x_2} = -\mu x_1^3 - \lambda x_2^3. \end{cases} \tag{4}$$

In this numerical experiment, we set parameters $\lambda = 0.1$ and $\mu = 2$, the simulation time interval $T = [0, 25]$, the step size $dt = 0.01$, the initial condition $X(0) = [2, 0]^\top$, the noise standard deviation $\epsilon = 0.05$ and the order of the polynomial basis library $d = 5$.

The results are summarized in three figures and one table, which collectively demonstrate the performance of ABSS-SINDy in identifying the system dynamics. Figure 6 shows the changes in objective value and modified relative error with the backward stepwise selection. Figure 7 shows the absolute value of coefficients on the same axis. Figure 8 shows the quantitative comparison between the true and the identified systems. Table 3 compares the coefficients between the true and the identified systems.

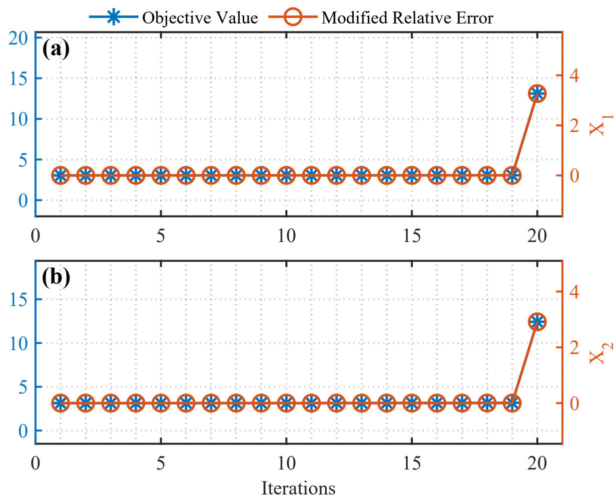
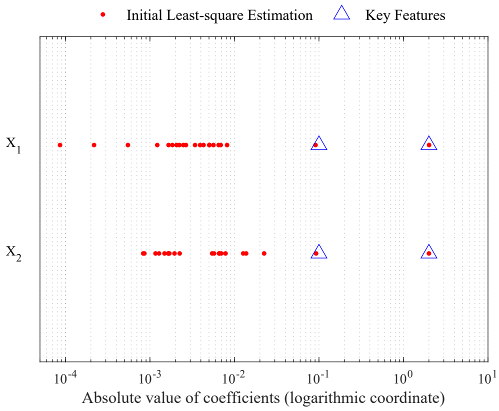
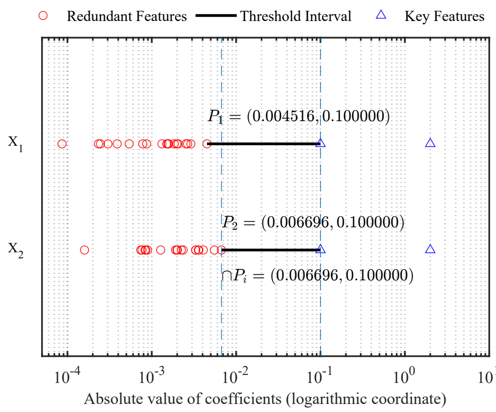


Figure 6: The process of the backward stepwise selection (positive example). The left y-axis represents the object value, and the right y-axis represents the MRE. Here, we construct the polynomial basis with the maximum order $d = 5$, and due to the state variables $n = 2$, the number of basis functions $p = \binom{n+d}{d} = 21$. The number of key features are $[n_{kf}]_1 = 2$ and $[n_{kf}]_2 = 2$. The hyperparameter $k_{\max}$ defined here will be accurately equal to the number of total features minus the number of key features: (a) $[k_{\max}]_1 = 19 = p - [n_{kf}]_1$; and (b) $[k_{\max}]_2 = 19 = p - [n_{kf}]_2$.



(a) Initial estimation based on least-square



(b) Final estimation based on backward stepwise selection

Figure 7: The absolute value of coefficients on the same axis (positive example). a) The red dots represent the initial estimated coefficients based on least squares. The blue triangles represent the key features of the system. b) The red circles represent the redundant features because the features will be deleted during the process of the backward stepwise selection. The blue triangles represent the key features of the system.

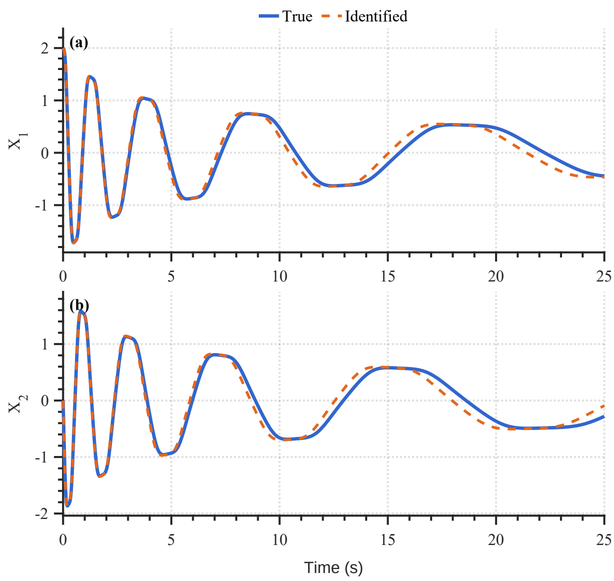


Figure 8: The quantitative comparison, measured by RMSE, between the true and the identified systems: (a) RMSE= 0.120044; and (b) RMSE = 0.113699.

Table 3: The Identified Cubic (2D) System

Structures	**Coefficients (True)**		**Coefficients (Identified)**	
	$\dot{\mathbf{x}}_1$	$\dot{\mathbf{x}}_2$	$\dot{\hat{\mathbf{x}}}_1$	$\dot{\hat{\mathbf{x}}}_2$
$x_1x_1x_1$	-0.1	-2	-0.099176	-2.000195
$x_2x_2x_2$	2	-0.1	1.999671	-0.097009

2.3 2D: The van der Pol Equation (positive example)

The van der Pol equation is a classical self-oscillation model, defined as follows:

$$\begin{cases} \dot{x_1} = x_2, \\ \dot{x_2} = -x_1 - \mu(x_1^2 - 1)x_2. \end{cases} \tag{5}$$

In this numerical experiment, we set parameter $\mu = 2$, the simulation time interval $T = [0, 25]$, the step size $dt = 0.01$, the initial condition $X(0) = [0.01, 0]^\top$, the noise standard deviation $\epsilon = 0.05$ and the order of the polynomial basis library $d = 2$.

The results are summarized in three figures and one table, which collectively demonstrate the performance of ABSS-SINDy in identifying the system dynamics. Figure 9 shows the changes in objective value and modified relative error with the backward stepwise selection. Figure 10 shows the absolute value of coefficients on the same axis. Figure 11 shows the quantitative comparison between the true and the identified systems. Table 4 compares the coefficients between the true and the identified systems.

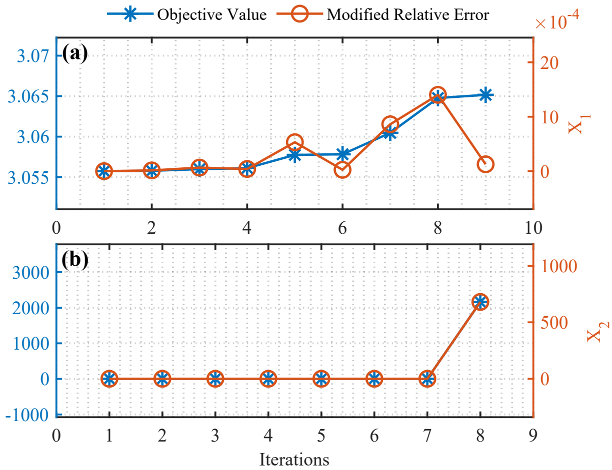


Figure 9: The process of the backward stepwise selection (positive example). The left y-axis represents the object value, and the right y-axis represents the MRE. Here, we construct the polynomial basis with the maximum order $d = 3$, and due to the state variables $n = 2$, the number of basis functions $p = \binom{n+d}{d} = 10$. The number of key features are $[n_{kf}]_1 = 1$ and $[n_{kf}]_2 = 3$. The hyperparameter $k_{\max}$ defined here will be accurately equal to the number of total features minus the number of key features: (a) $[k_{\max}]_1 = 9 = p - [n_{kf}]_1$; and (b) $[k_{\max}]_2 = 7 = p - [n_{kf}]_2$.

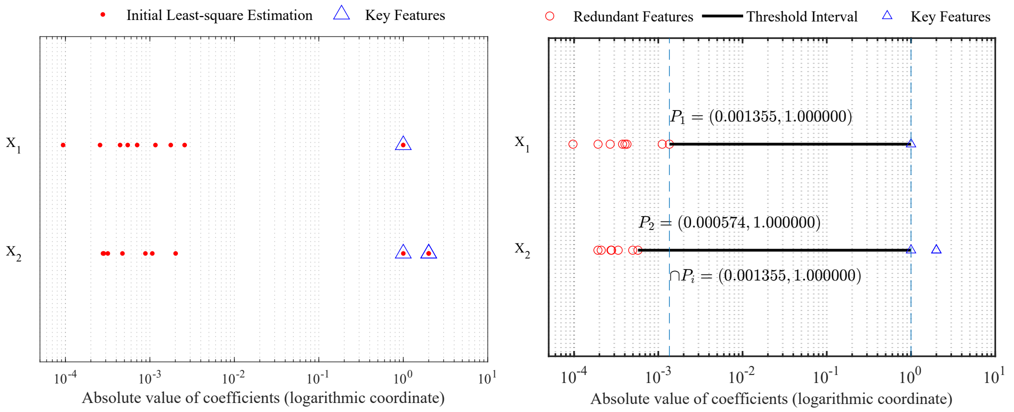
Table 4: The Identified van der Pol (2D) System

Structures	**Coefficients (True)**		**Coefficients (Identified)**	
	$\dot{\mathbf{x}}_1$	$\dot{\mathbf{x}}_2$	$\dot{\hat{\mathbf{x}}}_1$	$\dot{\hat{\mathbf{x}}}_2$
x_1	0	-1	0	-0.999289
x_2	1	2	0.999842	2.002351
$x_1x_1x_2$	0	-2	1.999671	-2.001726

2.4 3D: The Rössler equation (positive example)

The Rössler equation is a classical chaotic system, defined as follows:

$$\begin{cases} \dot{x_1} = -x_2 - x_3, \\ \dot{x_2} = x_1 + \alpha x_2, \\ \dot{x_2} = \alpha - \beta x_3 + x_1 x_3. \end{cases} \tag{6}$$



(a) Initial estimation based on least-square

(b) Final estimation based on backward stepwise selection

Figure 10: The absolute value of coefficients on the same axis (positive example). a) The red dots represent the initial estimated coefficients based on least squares. The blue triangles represent the key features of the system. b) The red circles represent the redundant features because the features will be deleted during the process of the backward stepwise selection. The blue triangles represent the key features of the system.

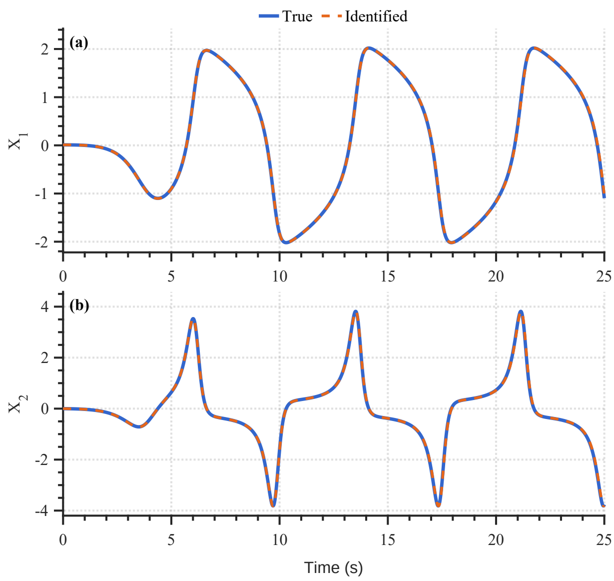


Figure 11: The quantitative comparison, measured by RMSE, between the true and the identified systems: (a) RMSE= 0.013530; and (b) RMSE = 0.026549.

In this numerical experiment, we set parameters $\alpha = 0.2$ and $\beta = 5.7$, the simulation time interval $T = [0, 50]$, the step size $dt = 0.01$, the initial condition $X(0) = [2.9, -1.3, 25]^\top$, the noise standard deviation $\epsilon = 1$ and the order of the polynomial basis library $d = 3$.

Due to the chaotic nature of the system, we set the significant level of separation between the two systems as 5% times the maximum absolute value of the state histories. The FPTs calculated are $\mathrm{FPT}_1 = 27.56$, $\mathrm{FPT}_2 = 28.97$, and $\mathrm{FPT}_3 = 21.45$. We can calculate the RMSE between the true and the identified systems in the finite prediction time interval $T = [0, 21.45]$.

The results are summarized in three figures and one table, which collectively demonstrate the performance of ABSS-SINDy in identifying the system dynamics. Figure 12 shows the changes in objective value and modified relative error with the backward stepwise selection. Figure 13 shows the absolute value of coefficients on the same axis. Figure 14 shows the quantitative comparison, captured by the RMSE in the given finite prediction time interval, between the true and the identified systems. Table 5 compares the coefficients between the true and the identified systems.

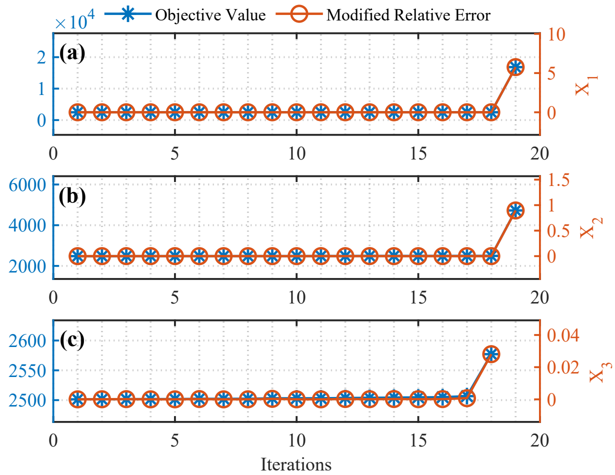
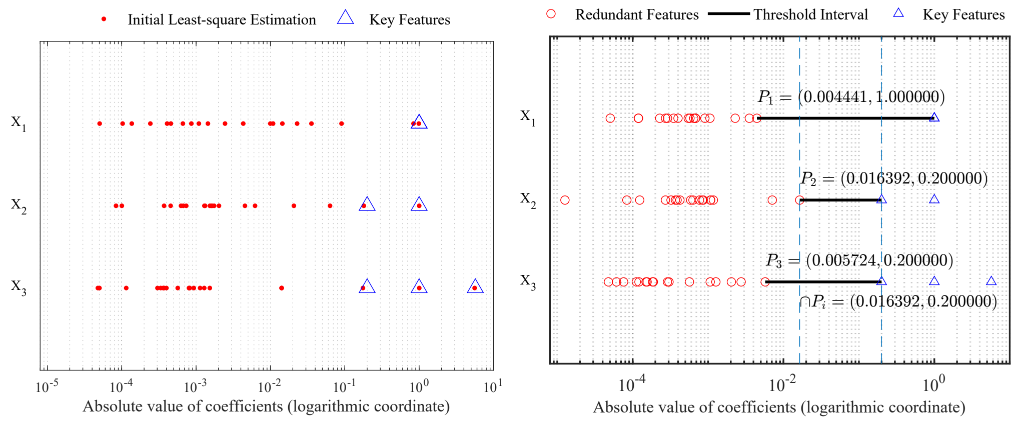


Figure 12: The process of the backward stepwise selection (positive example). The left y-axis represents the object value, and the right y-axis represents the MRE. Here, we construct the polynomial basis with the maximum order $d = 3$, and due to the state variables $n = 3$, the number of basis functions $p = \binom{n+d}{d} = 20$. The number of key features are $[n_{kf}]_1 = 2$, $[n_{kf}]_2 = 2$, and $[n_{kf}]_3 = 3$. The hyperparameter $k_{\max}$ defined here will be accurately equal to the number of total features minus the number of key features: (a) $[k_{\max}]_1 = 18 = p - [n_{kf}]_1$; (b) $[k_{\max}]_2 = 18 = p - [n_{kf}]_2$; and (c) $[k_{\max}]_3 = 17 = p - [n_{kf}]_3$.



(a) Initial estimation based on least-square

(b) Final estimation based on backward stepwise selection

Figure 13: The absolute value of coefficients on the same axis (positive example). a) The red dots represent the initial estimated coefficients based on least squares. The blue triangles represent the key features of the system. b) The red circles represent the redundant features because the features will be deleted during the process of the backward stepwise selection. The blue triangles represent the key features of the system.

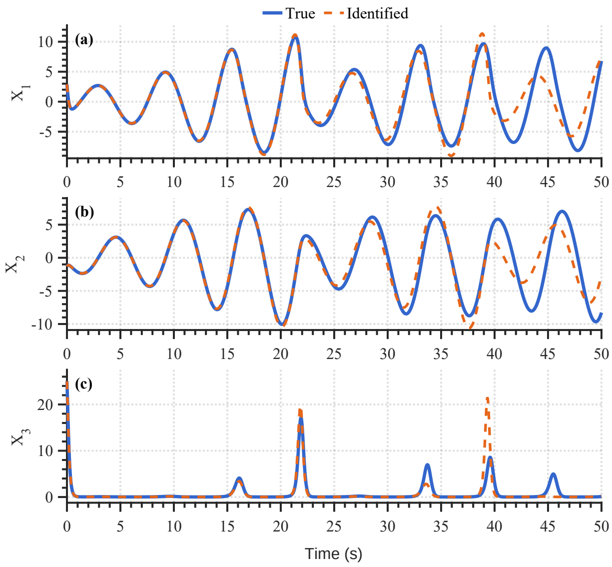


Figure 14: The quantitative comparison, measured by RMSE in $T = [0, 9.17]$, between the true and the identified systems: (a) RMSE= 0.036394; (b) RMSE = 0.051188; and (c) RMSE = 0.005319.

Table 5: The Identified Rössler (3D) System

Structures	**Coefficients (True)**			**Coefficients (Identified)**		
	$\dot{\mathbf{x}}_1$	$\dot{\mathbf{x}}_2$	$\dot{\mathbf{x}}_3$	$\dot{\hat{\mathbf{x}}}_1$	$\dot{\hat{\mathbf{x}}}_2$	$\dot{\hat{\mathbf{x}}}_3$
1	0	0	0.2	0	0	0.177034
x_1	0	1	0	0	1.000688	0
x_2	-1	0.2	0	-1.004537	0.197168	0
x_3	-1	0	-5.7	-1.000335	0	-5.694053
x_1x_3	0	0	1	0	0	0.999867

2.5 3D: The Lorenz 63 equation (positive example)

The Lorenz 63 equation is a classical chaotic system, defined as follows:

$$\begin{cases} \dot{x_1} = -\sigma(x_2 - x_1), \\ \dot{x_2} = x_1(\rho - x_3) - x_2, \\ \dot{x_2} = x_1x_2 - \beta x_3. \end{cases} \tag{7}$$

In this numerical experiment, we set parameters $\sigma = 10$, $\beta = 8/3$ and $\rho = 28$, the simulation time interval $T = [0, 50]$, the step size $dt = 0.01$, the initial condition $X(0) = [-8, 8, 27]^\top$, the noise standard deviation $\epsilon = 1$ and the order of the polynomial basis library $d = 3$.

Due to the chaotic nature of the system, we set the significant level of separation between the two systems as 5% times the maximum absolute value of the state histories. The FPTs calculated are $\text{FPT}_1 = 5.82$, $\text{FPT}_2 = 5.58$, and $\text{FPT}_3 = 5.62$. We can calculate the RMSE between the true and the identified systems in the finite prediction time interval $T = [0, 5.59]$.

The results are summarized in three figures and one table, which collectively demonstrate the performance of ABSS-SINDy in identifying the system dynamics. Figure 15 shows the changes in objective value and modified relative error with the backward stepwise selection. Figure 16 shows the absolute value of coefficients on the same axis. Figure 17 shows the quantitative comparison, captured by the RMSE in the given finite prediction time interval, between the true and the identified systems. Table 6 compares the coefficients between the true and the identified systems.

Table 6: The Identified Lorenz (3D) System

Structures	**Coefficients (True)**			**Coefficients (Identified)**		
	$\dot{\mathbf{x}}_1$	$\dot{\mathbf{x}}_2$	$\dot{\mathbf{x}}_3$	$\dot{\hat{\mathbf{x}}}_1$	$\dot{\hat{\mathbf{x}}}_2$	$\dot{\hat{\mathbf{x}}}_3$
x_1	10	28	0	-9.996421	27.988079	0
x_2	-10	-1	0	10.000322	-0.997950	0
x_3	0	0	$-8/3$	0	0	-2.665999
x_1x_2	0	0	1	0	0	0.999605
x_1x_3	0	-1	0	0	-0.999722	0

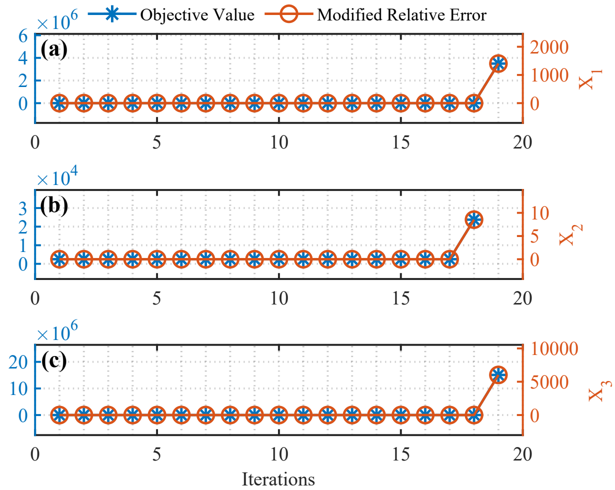
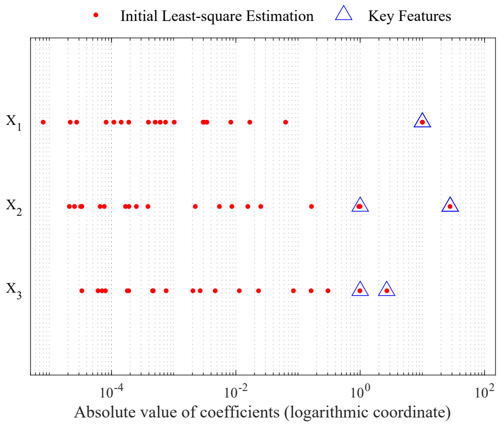
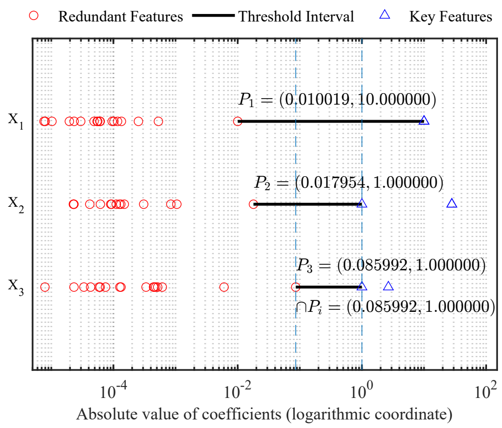


Figure 15: The process of the backward stepwise selection (positive example). The left y-axis represents the object value, and the right y-axis represents the MRE. Here, we construct the polynomial basis with the maximum order $d = 3$, and due to the state variables $n = 3$, the number of basis functions $p = \binom{n+d}{d} = 20$. The number of key features are $[n_{kf}]_1 = 2$, $[n_{kf}]_2 = 3$, and $[n_{kf}]_3 = 2$. The hyperparameter $k_{\max}$ defined here will be accurately equal to the number of total features minus the number of key features: (a) $[k_{\max}]_1 = 18 = p - [n_{kf}]_1$; (b) $[k_{\max}]_2 = 17 = p - [n_{kf}]_2$; and (c) $[k_{\max}]_3 = 18 = p - [n_{kf}]_3$.



(a) Initial estimation based on least-square



(b) Final estimation based on backward stepwise selection

Figure 16: The absolute value of coefficients on the same axis (positive example). a) The red dots represent the initial estimated coefficients based on least-square. The blue triangles represent the key features of the system. b) The red circles represent the redundant features because the features will be deleted during the process of the backward stepwise selection. The blue triangles represent the key features of the system.

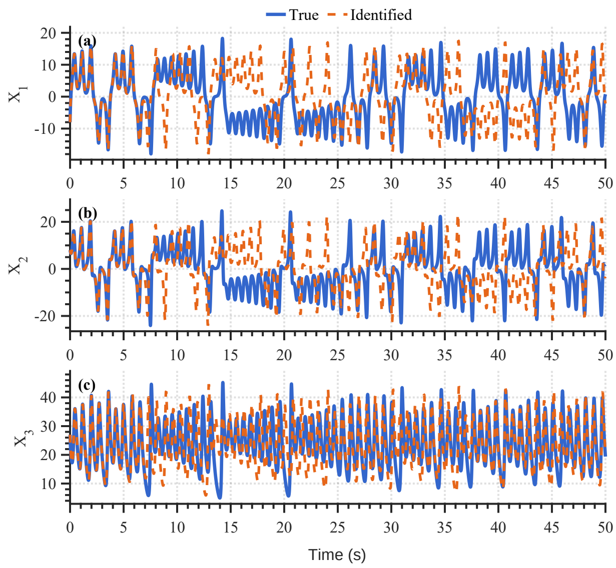


Figure 17: The quantitative comparison, measured by RMSE in $T = [0, 5.59]$, between the true system and the identified system: (a) RMSE= 0.198555; (b) RMSE = 0.348647; and (c) RMSE = 0.287923.

2.6 3D: The linear equation (counterexample)

$$\begin{cases} \dot{x_1} = -\lambda x_1 + \mu x_2, \\ \dot{x_2} = -\mu x_1 - \lambda x_2, \\ \dot{x_3} = -\rho x_3. \end{cases} \tag{8}$$

In this numerical experiment, we set parameters $\lambda = 0.0001$, $\mu = 0.002$ and $\rho = 0.0003$, the simulation time interval $T = [0, 5000]$, the step size $dt = 0.05$, the initial condition $X(0) = [2, 0, 1]^\top$, the noise standard deviation $\epsilon = 0.0001$ and the order of the polynomial basis library $d = 2$.

The results are summarized in three figures and one table, which collectively demonstrate the performance of ABSS-SINDy in identifying the system dynamics. Figure 18 shows the changes in objective value and modified relative error with the backward stepwise selection. Figure 19 shows the quantitative comparison, captured by the RMSE, between the true and the identified systems. Table 7 compares the coefficients between the true and the identified systems.

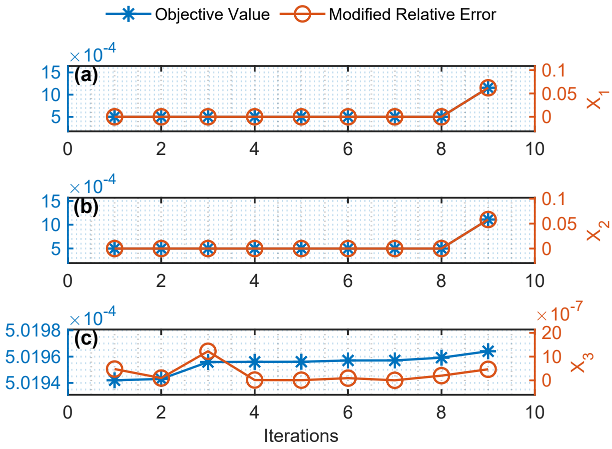


Figure 18: The process of the backward stepwise selection (counterexample). The left y-axis represents the object value, and the right y-axis represents the MRE. Here, we construct the polynomial basis with the maximum order $d = 2$, and due to the state variables $n = 3$, the number of basis functions $p = \binom{n+d}{d} = 10$. The number of key features are $[n_{kf}]_1 = 2$, $[n_{kf}]_2 = 2$, and $[n_{kf}]_3 = 1$. The hyperparameter $k_{\max}$ defined here will be accurately equal to the number of total features minus the number of key features: (a) $[k_{\max}]_1 = 8 = p - [n_{kf}]_1$; (b) $[k_{\max}]_2 = 8 = p - [n_{kf}]_2$; and (c) $[k_{\max}]_3 = 9 = p - [n_{kf}]_3$.

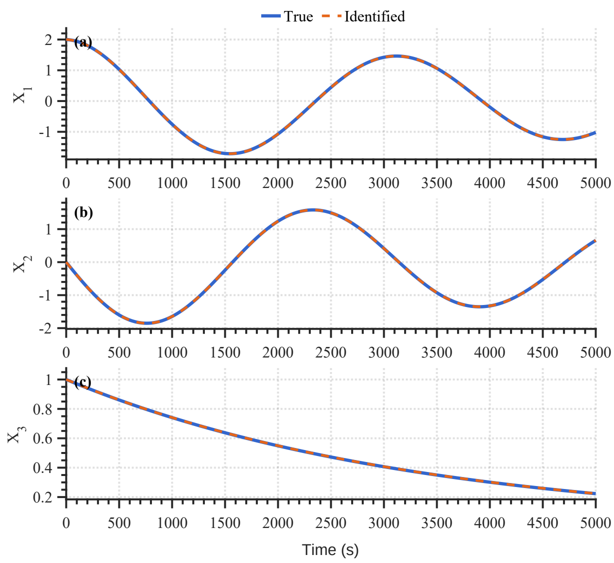


Figure 19: The quantitative comparison, measured by RMSE, between the true system and the identified system: (a) RMSE= 0.000749; (b) RMSE = 0.000682; and (c) RMSE = 0.000557.

Table 7: The Identified Linear (3D) System

Structures	Coefficients (True)			Coefficients (Identified)		
	$\dot{\mathbf{x}}_1$	$\dot{\mathbf{x}}_2$	$\dot{\mathbf{x}}_3$	$\hat{\dot{\mathbf{x}}}_1$	$\hat{\dot{\mathbf{x}}}_2$	$\hat{\dot{\mathbf{x}}}_3$
x_1	−0.000100	−0.002000	0	−0.000100	−0.002000	0
x_2	0.002000	−0.000100	0	0.002000	−0.000100	0
x_3	0	0	−0.000300	0	0	−0.000299

2.7 4D: The Lotka-Volterra (positive example)

$$\begin{cases} \dot{x} = x(1 - x - 2y - 1.3z), \\ \dot{y} = y(1 - y - z - w), \\ \dot{z} = z(1 - 2x - z), \\ \dot{w} = w(1 - x - w). \end{cases} \tag{9}$$

In this numerical experiment, the parameters are embedded in the equations, and we set the simulation time interval $T = [0, 500]$, the step size $dt = 0.01$, the initial condition $X(0) = [0.5, 0.2, 0.1, 0.5]^\top$, the noise standard deviation $\epsilon = 0.001$ and the order of the polynomial basis library $d = 2$.

The results are summarized in three figures and one table, which collectively demonstrate the performance of ABSS-SINDy in identifying the system dynamics. Figure 20 shows the changes in objective value and modified relative error with the backward stepwise selection. Figure 21 shows the quantitative comparison, captured by the RMSE, between the true and the identified systems. Table 7 compares the coefficients between the true and the identified systems.

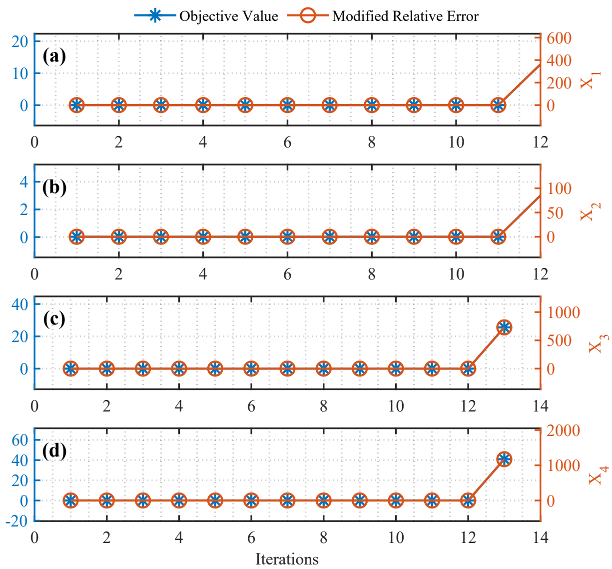


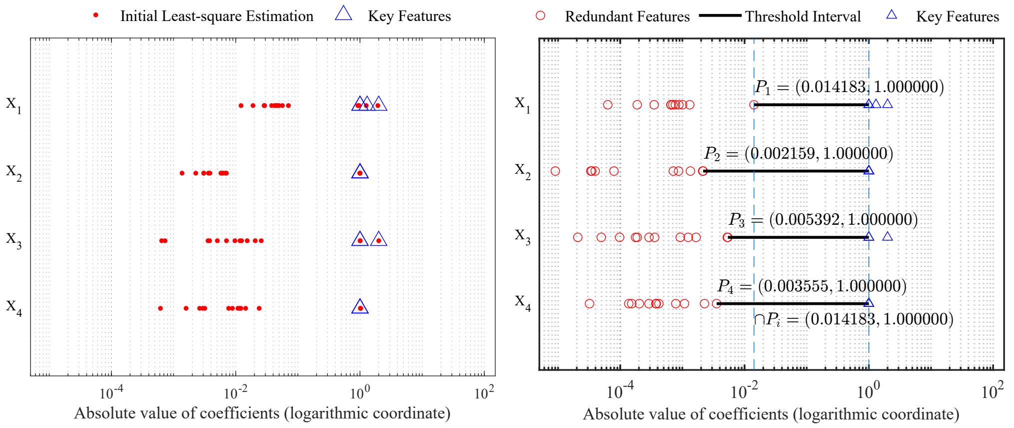
Figure 20: The process of the backward stepwise selection (positive example). The left y-axis represents the object value, and the right y-axis represents the MRE. Here, we construct the polynomial basis with the maximum order $d = 2$, and due to the state variables $n = 4$, the number of basis functions $p = \binom{n+d}{d} = 15$. The number of key features are $[n_{kf}]_1 = 4$, $[n_{kf}]_2 = 4$, $[n_{kf}]_3 = 3$, and $[n_{kf}]_4 = 3$. The hyperparameter $k_{\max}$ defined here will be accurately equal to the number of total features minus the number of key features: (a) $[k_{\max}]_1 = 11 = p - [n_{kf}]_1$; (b) $[k_{\max}]_2 = 11 = p - [n_{kf}]_2$; (c) $[k_{\max}]_3 = 12 = p - [n_{kf}]_3$; and (d) $[k_{\max}]_4 = 12 = p - [n_{kf}]_4$.

2.8 4D: The Rössler Hyperchaotic (positive example)

$$\begin{cases} \dot{x} = -y - z, \\ \dot{y} = x + ay + w, \\ \dot{z} = b + xz, \\ \dot{w} = cw + dz. \end{cases} \tag{10}$$

In this numerical experiment, we set parameters $a = 0.25$, $b = 3$, $c = 0.05$, and $d = 0.5$, the simulation time interval $T = [0, 200]$, the step size $dt = 0.01$, the initial condition $X(0) = [2, 0, 1]^\top$, the noise standard deviation $\epsilon = 0.005$ and the order of the polynomial basis library $d = 2$.

The results are presented in three figures, which demonstrate the performance of ABSS-SINDy with respect to the identified system dynamics. Figure 23 shows the changes in objective value and modified relative error with the backward stepwise selection. Figure 24 shows the absolute value of coefficients on the same axis. Figure 25 shows the quantitative comparison between the true and the identified systems.



(a) Initial estimation based on least-square

(b) Final estimation based on backward stepwise selection

Figure 21: The absolute value of coefficients on the same axis (positive example). (a) The red dots represent the initial estimated coefficients based on least squares. The blue triangles represent the key features of the system. (b) The red circles represent the redundant features because the features will be deleted during the process of the backward stepwise selection. The blue triangles represent the key features of the system.

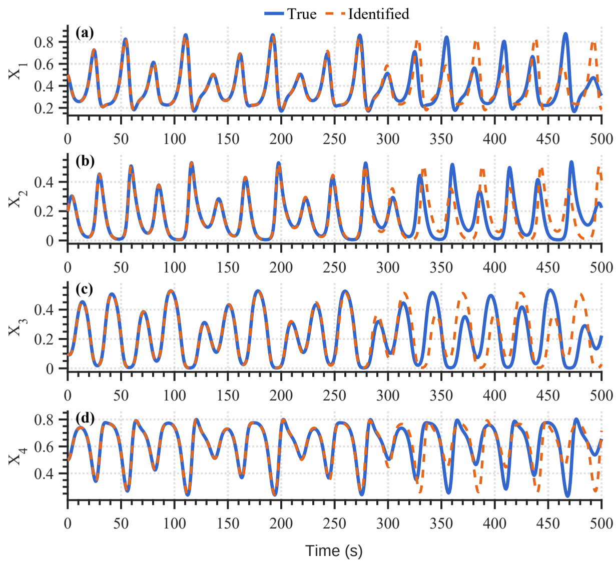


Figure 22: The quantitative comparison, measured by RMSE in $T = [0, 274.47]$, between the true system and the identified system: (a) RMSE= 0.008308; (b) RMSE = 0.007600; (c) RMSE = 0.008658; and (d) RMSE = 0.006601.

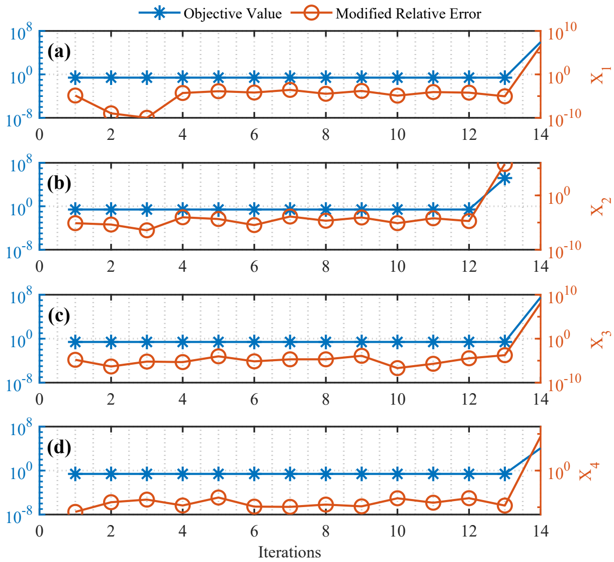
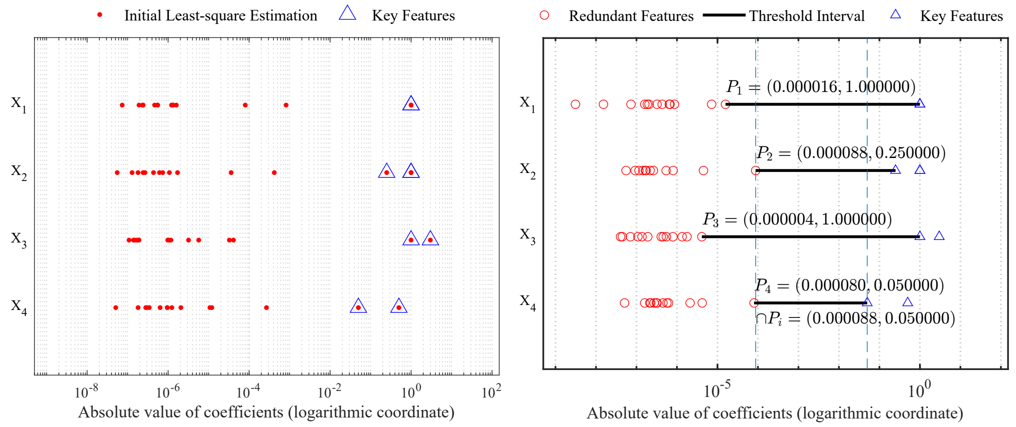


Figure 23: The process of the backward stepwise selection (positive example). The left y-axis represents the object value, and the right y-axis represents the MRE. Here, we construct the polynomial basis with the maximum order $d = 2$, and due to the state variables $n = 4$, the number of basis functions $p = \binom{n+d}{d} = 15$. The number of key features are $[n_{kf}]_1 = 2$, $[n_{kf}]_2 = 3$, $[n_{kf}]_3 = 2$, and $[n_{kf}]_4 = 2$. The hyperparameter $k_{\max}$ defined here will be accurately equal to the number of total features minus the number of key features: (a) $[k_{\max}]_1 = 13 = p - [n_{kf}]_1$; (b) $[k_{\max}]_2 = 12 = p - [n_{kf}]_2$; (c) $[k_{\max}]_3 = 13 = p - [n_{kf}]_3$; and (d) $[k_{\max}]_4 = 13 = p - [n_{kf}]_4$.



(a) Initial estimation based on least-square

(b) Final estimation based on backward stepwise selection

Figure 24: The absolute value of coefficients on the same axis (positive example). (a) The red dots represent the initial estimated coefficients based on least squares. The blue triangles represent the key features of the system. (b) The red circles represent the redundant features because the features will be deleted during the process of the backward stepwise selection. The blue triangles represent the key features of the system.

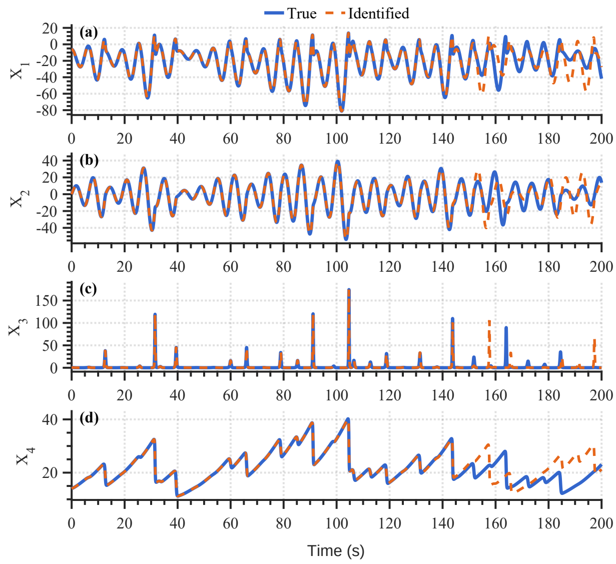


Figure 25: The quantitative comparison, measured by RMSE in $T = [0, 143.64]$, between the true system and the identified system: (a) RMSE= 0.196708; (b) RMSE = 0.186081; (c) RMSE = 0.297946; and (d) RMSE = 0.064423.

2.9 6D: The coupled Lorenz system (positive example)

$$\begin{cases} \dot{x} = -x + y + ku, \\ \dot{y} = -xz, \\ \dot{z} = -R + xy, \\ \dot{u} = -u + v + kx, \\ \dot{v} = -uw, \\ \dot{w} = uv - R. \end{cases} \tag{11}$$

In this numerical experiment, we set parameters $k = 0.5$ and $R = 1$, the simulation time interval $T = [0, 200]$, the step size $dt = 0.01$, the initial condition $X(0) = [0, 0.8, 0.4, 0, 1, 0]^\top$, the noise standard deviation $\epsilon = 0.005$ and the order of the polynomial basis library $d = 2$.

The results are presented in three figures, which demonstrate the performance of ABSS-SINDy with respect to the identified system dynamics. Figure 26 shows the changes in objective value and modified relative error with the backward stepwise selection. Figure 27 shows the absolute value of coefficients on the same axis. Figure 28 shows the quantitative comparison between the true and the identified systems.

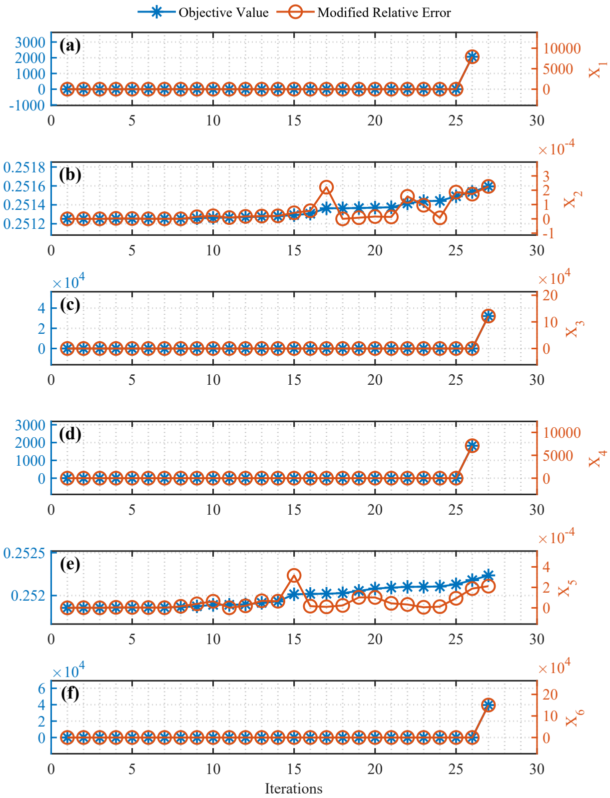
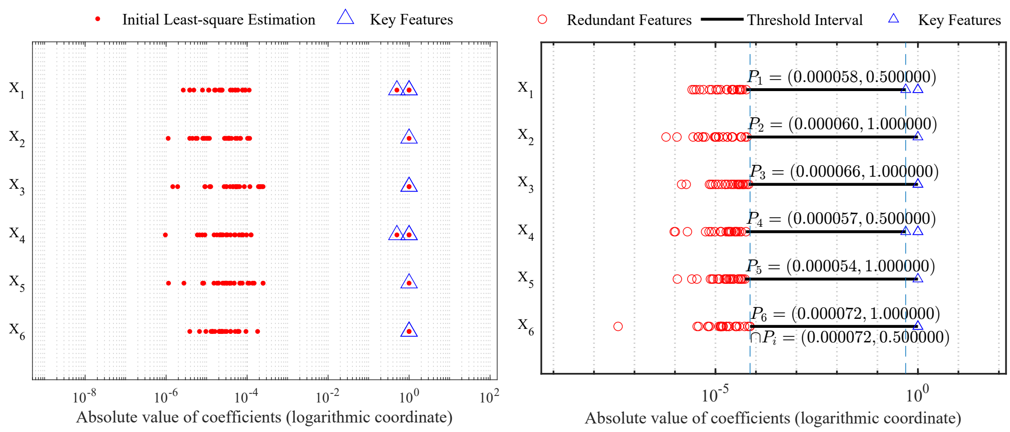


Figure 26: The process of the backward stepwise selection (positive example). The left y-axis represents the object value, and the right y-axis represents the MRE. Here, we construct the polynomial basis with the maximum order $d = 2$, and due to the state variables $n = 6$, the number of basis functions $p = \binom{n+d}{d} = 28$. The number of key features are $[n_{kf}]_1 = 3$, $[n_{kf}]_2 = 1$, $[n_{kf}]_3 = 2$, $[n_{kf}]_4 = 3$, $[n_{kf}]_5 = 1$, $[n_{kf}]_6 = 2$. The hyperparameter $k_{\max}$ defined here will be accurately equal to the number of total features minus the number of key features: (a) $[k_{\max}]_1 = 25 = p - [n_{kf}]_1$; (b) $[k_{\max}]_2 = 27 = p - [n_{kf}]_2$; (c) $[k_{\max}]_3 = 26 = p - [n_{kf}]_3$; (d) $[k_{\max}]_1 = 25 = p - [n_{kf}]_1$; (e) $[k_{\max}]_2 = 27 = p - [n_{kf}]_2$; (f) $[k_{\max}]_3 = 26 = p - [n_{kf}]_3$.



(a) Initial estimation based on least-square

(b) Final estimation based on backward stepwise selection

Figure 27: The absolute value of coefficients on the same axis (positive example). (a) The red dots represent the initial estimated coefficients based on least squares. The blue triangles represent the key features of the system. (b) The red circles represent the redundant features because the features will be deleted during the process of the backward stepwise selection. The blue triangles represent the key features of the system.

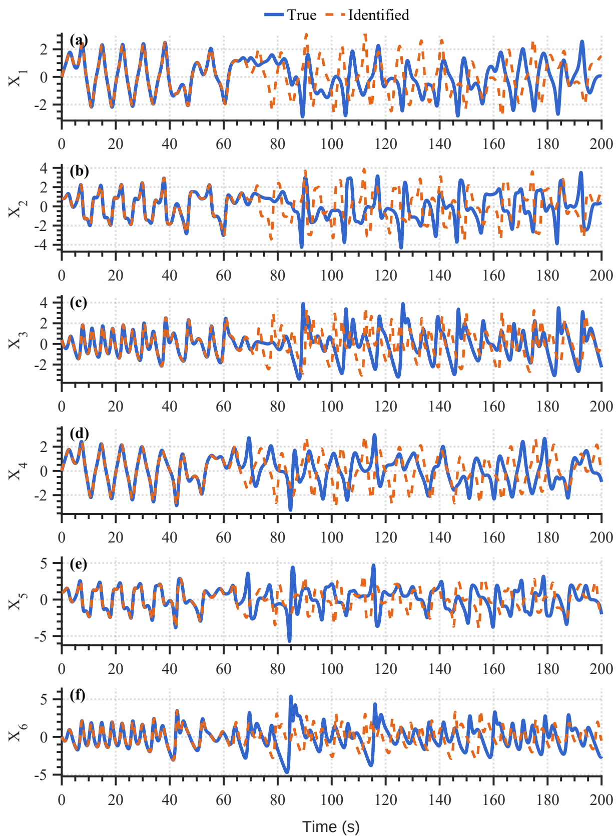


Figure 28: The quantitative comparison, measured by RMSE in $T = [0, 59.59]$, between the true system and the identified system: (a) RMSE= 0.017577; (b) RMSE = 0.025767; (c) RMSE = 0.031038; (d) RMSE = 0.017259; (e) RMSE = 0.027864; and (f) RMSE = 0.036715.

3 Justification for parameter selection

Some readers found that we introduce extra hyperparameters, the regularization parameter η in Eq. 12 and the modification factor γ for the relative error in Eq. 12, in the manuscript. Is it reasonable to set those parameters as fixed values? How can we analyze the robustness and the stability according to practical cases?

$$\begin{cases} \xi^{(k+1)} \longleftarrow \arg\min_{\xi} \left\{ \|\mathbf{\Theta}(\boldsymbol{X})\xi - \mathrm{d}X\|_2^2 + \eta\|\xi\|^2 \quad \text{s.t. } \mathcal{S}(\xi) = \mathcal{S}\left(\xi^{(k)}\right) \right\}, \\ \lambda^{(k+1)} \longleftarrow \min \left\{ |\xi^{(k+1)}(i)| : \xi^{(k+1)}(i) \neq 0 \right\}, \\ \xi^{(k+1)} \longleftarrow \mathcal{T}\left(\xi^{(k+1)}; \lambda^{(k+1)}\right), \end{cases} \tag{12}$$

$$\mathrm{MRE}(x, \hat{x}, \gamma) = \frac{|x - \hat{x}|}{\gamma + |x|}, \tag{13}$$

We must admit that the fixed values are not always valid for all problems. After all, we can always find extreme cases or construct specific examples. Regarding the above two questions, we have stated two comments in the manuscript, as follows,

- The regularization parameter may vary across other problems. Section 2.2 (in the manuscript) provides a quantitative analysis for the ridge regression, where a more reasonable parameter selection can be determined according to the 2-norm condition number. Moreover, the stability of the solution can be quantitatively evaluated.
- From another point of view, the parameter γ can be determined more precisely, especially when the fixed value fails for other problems. For the given x and $\hat{x}$ of MRE, the MRE is only dependent on the fixed value of γ. Thus, MRE is a function of γ. The function graph can be drawn as γ changes, which will enable a more reasonable choice and sensitivity analysis of the parameter.

The two comments will allow us to set more reasonable parameters according to the properties of practical problems. Throughout the study, we set them as fixed values, where $\eta = 10^{-6}$ and $\gamma = 0.01$. We conduct a series of numerical experiments involving both planar/spatial dynamics and high-dimensional chaotic systems, including Lotka-Volterra, Hyperchaotic Rössler, Coupled Lorenz, and Lorenz 96 benchmark systems. All the results strengthened our belief in the fixed values of parameters. From this perspective, perhaps we should not consider the above two parameters as hyperparameters.

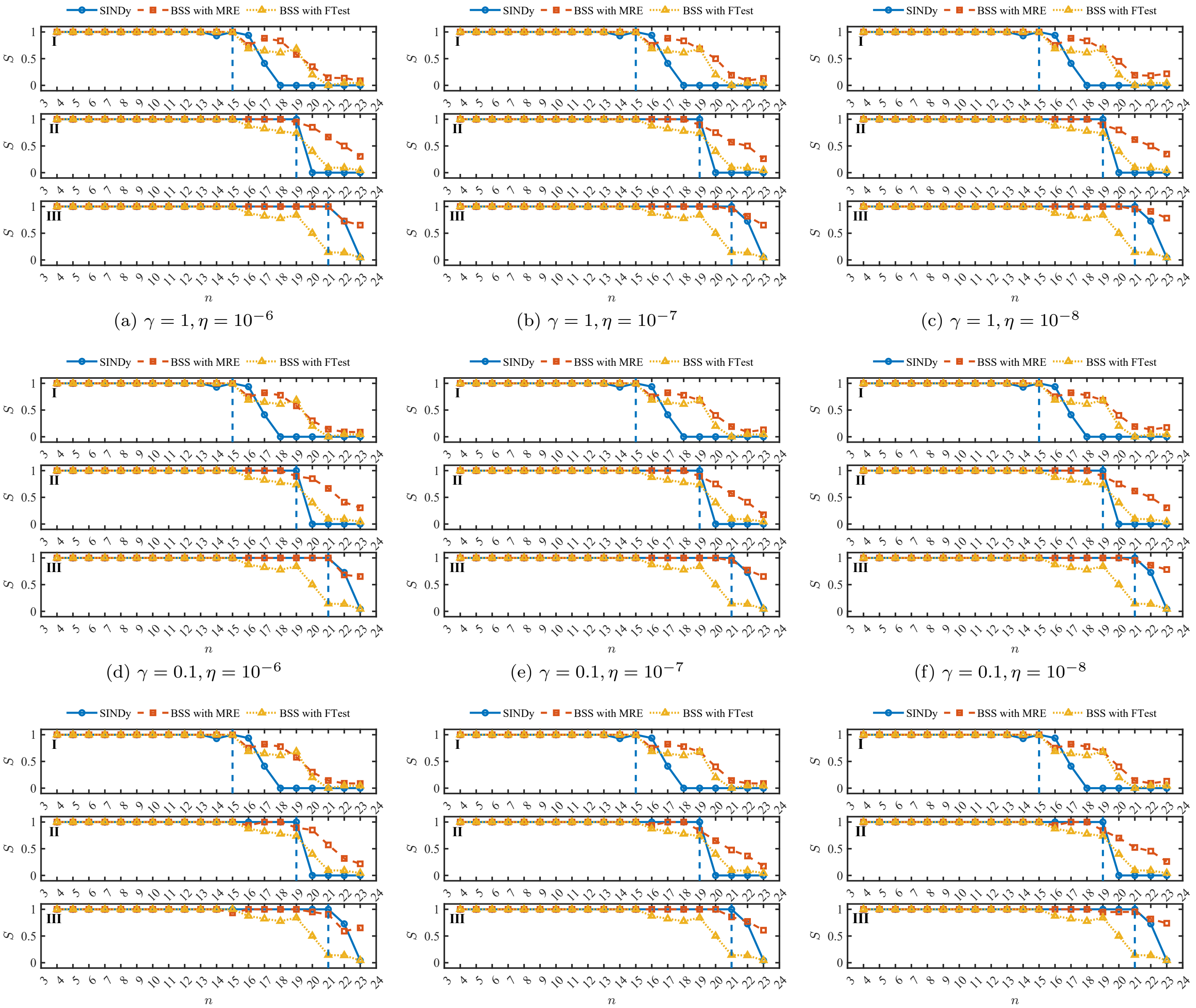
Generally speaking, the hyperparameters tend to vary significantly depending on the specific problem. Here, we take the well-known original SINDy method as an example. In that algorithm, the threshold value λ is highly dependent on the problems. However, the maximum iteration $k_{\max}$ is set to a fixed value to ensure that sufficient iterations are carried out to filter out all values below the threshold value. From this point, the threshold value λ is far more important and sensitive than the maximum iteration $k_{\max}$. The threshold value λ should be considered as a standard example of hyperparameter. However, the maximum iteration $k_{\max}$ should be slightly changed in high-dimensional cases.

Based on the above qualitative analysis, we can conclude that the two parameters used in this study are robust and insensitive to the problems. We also perform some experiments to support our viewpoint. In this numerical experiment and following the same Lorenz 96 model as shown in the manuscript, we set the parameters $n = 4, 5, \dots, 23$, $F = 8$, the simulation time interval $T = [0, 15]$, the step size $dt = 0.01$, the initial condition $X(0) = [1, 8, 8, \dots, 8]^\top$, and the order of the polynomial basis library $d = 2$. This system with variable dimensions can fully demonstrate our results.

Some important parameters are set as follows. The noise standard deviation ϵ is set to represent three cases, Case I: $\epsilon = 1$, Case II: $\epsilon = 0.1$, and Case III: $\epsilon = 0.01$, following the same situation as shown in the manuscript. The regularization parameter η is set to $\eta = 10^{-6}, 10^{-7}, \text{and } 10^{-8}$. The modification factor γ is set to $\gamma = 1, 0.1, \text{and } 0.01$. These parameter settings allow us to explore a large parameter space, which will enable us to analyze the robustness and sensitivity.

The detailed comparative results are shown in Fig.,29, which further strengthens our claim stated in the main text, as follows,

- "This result illustrates that while SINDy fails abruptly as the system dimension increases, the F-test and MRE can still capture some correct structures. In this case, MRE performs at a similar level to the F-test. However, as the noise standard deviation decreases, the difference between the two error metrics becomes more significant. These numerical experiments verify the limitations of the F-test, as analyzed in Sec. 2, and also reveal some limitations of our proposed method. Nevertheless, further examples, both in subsequent sections and in the Supporting Information, will demonstrate the effectiveness of MRE."



(a) $\gamma = 1, \eta = 10^{-6}$ (b) $\gamma = 1, \eta = 10^{-7}$ (c) $\gamma = 1, \eta = 10^{-8}$

(d) $\gamma = 0.1, \eta = 10^{-6}$ (e) $\gamma = 0.1, \eta = 10^{-7}$ (f) $\gamma = 0.1, \eta = 10^{-8}$

(g) $\boldsymbol{\gamma = 0.01, \eta = 10^{-6}}$ (h) $\gamma = 0.01, \eta = 10^{-7}$ (i) $\gamma = 0.01, \eta = 10^{-8}$

Figure 29: The results throughout different noise standards, regularization parameters, and modification factors are shown. The subplot (g) represents the result in our manuscript.