

Discover governing differential equations from evolving systems

Yuanyuan Li¹, Kai Wu^{2,*}, and Jing Liu¹

¹Guangzhou Institute of Technology, Xidian University, Guangzhou 510555, China

²School of Artificial Intelligence, Xidian University, Xi'an 710071, China



(Received 2 January 2023; accepted 9 May 2023; published 26 May 2023)

Discovering the governing equations of evolving systems from available observations is essential and challenging. In this paper, we consider a different scenario: discovering governing equations from streaming data. Current methods struggle to discover governing differential equations with considering measurements as a whole, leading to failure to handle this task. We propose an online modeling method capable of handling samples one by one sequentially by modeling streaming data instead of processing the entire data set. The proposed method performs well in discovering ordinary differential equations and partial differential equations from streaming data. Evolving systems are changing over time, which invariably changes with system status. Thus finding the exact change points is critical. The measurement generated from a changed system is distributed dissimilarly to before; hence the difference can be identified by the proposed method. Our proposal is competitive in identifying the change points and discovering governing differential equations in three hybrid systems and two switching linear systems.

DOI: [10.1103/PhysRevResearch.5.023126](https://doi.org/10.1103/PhysRevResearch.5.023126)

I. INTRODUCTION

Research on the hidden information in measurements that performs a transformative impact on the discovery of dynamical systems has attracted the attention of scientific researchers [1–4]. Many canonical dynamics are rooted in conservation laws and phenomenological behaviors in physical engineering and biological science [5]. Enabled by the increasing maturity of sensor technology, data-driven methods have promoted various innovations in describing time series generated from experiments [4]. Due to the complexity of dynamical systems, revealing the underlying governing equations representing the physical laws from time-series data that gives a general description of the spatiotemporal activities is a tremendous challenge [3,6–12].

In a series of developments, modeling methods for complex systems include empirical dynamic modeling [13], equation-free modeling [14,15], and modeling emergent behavior [16]. Discovery patterns contain normal form identification [17], nonlinear Laplacian spectral analysis [18], dynamic automatic inference [19], and nonlinear regression [20]. Overall, popular methods for data-driven discovery of complex systems are mainly based on sparse identification [21,22] and deep neural networks [23,24], such as sparse identification of nonlinear dynamics (SINDy) [25] and PDE-Net [26]. Existing methods have provided an encouraging

performance on systems motivated by unchanging rules or architectures, where novelty and variability are seen as disruptive [27].

However, although capable of effectively handling spatiotemporal sequences, the above algorithms are batch learning methodologies whose common problem is ephemeral fitting [13,28], thereby leading to the abortive modeling for streaming data consecutively generated like water flow. The overwhelming majority of complex systems are constantly running and generating new observations one by one over time (such as financial markets, social networks, neurological connectivity, etc.) [29–32], resulting in failing to update gradients in the SINDy. In real-time systems, we must adapt to different stream parts and produce an immediate output before seeing the next input.

Moreover, several SINDy-based methods have been demonstrated to perform online tasks. Autoencoder SINDy is trained offline and queried online to compute the time evolution of the full order system in different conditions [33]. Reference [34] claims that the SINDy-based algorithm holds feasibility in online applications due to its little computational time for all cases, owing to a little tuning of parameters and fast Fourier transform. However, these methods cannot handle streaming data and discover the change points in evolving systems. The most related work is the method in [35], which processes solution snapshots that arrive sequentially. It is a small batch method. Its performance and training iterations depend on the length of snapshots, which will seriously affect the evaluation of change points. Thus Ref. [35] has difficulty finding the change point hidden in solution snapshots on account of the nature of the minibatch method, which limits the application of this method to switching systems.

To the concrete, we regard the measurements reflecting the system status as invariably changing streaming data. In this

*kwu@xidian.edu.cn

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

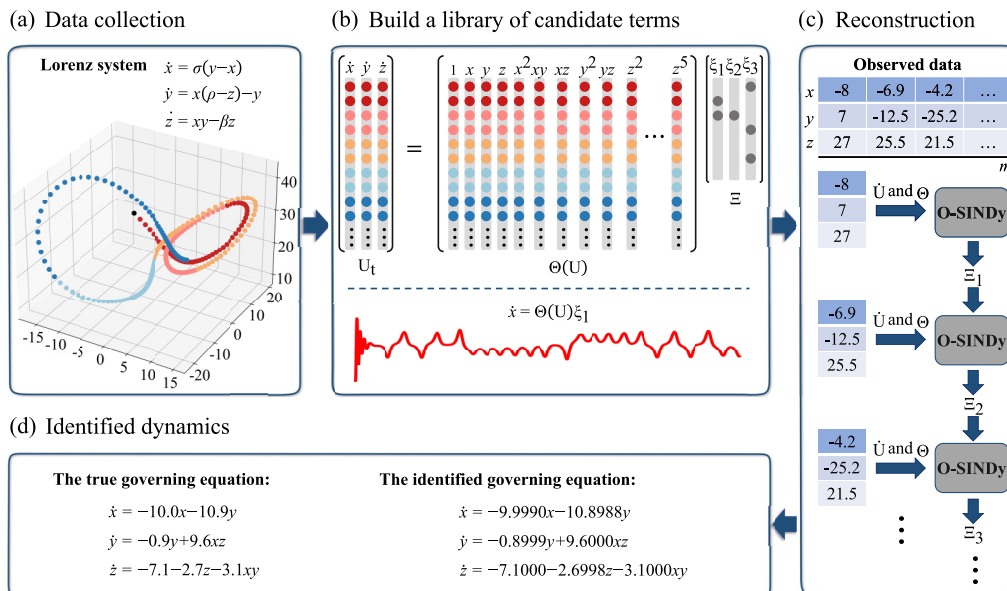


FIG. 1. Steps in O-SINDy applied to infer the governing equations of the chaotic Lorenz system from streaming data. (a) The particle’s trajectory of motion, starting from the black point. The points of different colors represent the record collected as streams in real-time systems but used as a whole by batch methods. (b) As the observations arrive, take numerical derivatives of the current state vector and construct a library matrix Θ incorporating the candidate terms for governing equations. (c) Update the form of equations by considering one instance each time. (d) The reconstruction of the Lorenz system.

article, we present a method, called online-sparse identification of nonlinear dynamics (O-SINDy), to develop a parsimonious model that most accurately represents the real-time streaming data. Our proposal handles samples sequentially by modeling subsampled spatiotemporal sequences as streaming measurements instead of processing the entire data set directly. Moreover, O-SINDy can bypass batch storage and intractable cases of combinatorial brute-force search across all possible candidate terms.

Mathematically, we conduct experiments with diverse spatiotemporal measurements generated from dynamical systems to verify the excellent performance of O-SINDy. By recovering extensively representative physical system expressions solely from streaming data, the method is demonstrated to work on various canonical instances, including the nonlinear Lorenz system, Burgers’ equation, reaction-diffusion equation, etc. Experimental results show that our framework provides an online technique for real-time online analysis of complex systems. We suggest that O-SINDy, which overcomes the limitations of batch learning methodologies, is competent for deducing the governing differential equations if sequential measurements of complex dynamic systems are available.

With the ability to handle streaming data, O-SINDy can ideally cope with the parameter estimation of time-varying nonlinear dynamics and is general enough to detect multiple types of evolutionary patterns. O-SINDy outperforms the state-of-the-art methods in discovering governing differential equations in the evolving two-dimensional damped harmonic oscillators, susceptible-infected-recovered disease model with time-varying transmission rates, and two switching linear systems.

II. METHODS

With large amounts of data arriving as streams, any offline machine learning algorithm that attempts to store the complete data set for analysis will fail due to running out of memory. The rise of streaming data raises the technical challenge of a real-time system, which must do any processing or learning encouraged by each record in arriving order. Let u denote the state vector of a real-time system at time t and u_t represent its partial differentiation in time domain. The system receives streaming inputs:

$$\dots, u_{t-3}, u_{t-2}, u_{t-1}, u_t, u_{t+1}, u_{t+2}, u_{t+3}, \dots$$

For instance, vector u represents the position information of the particle shown in Fig. 1(a). Every step, the particle moves along the trajectory; the data will be captured by sensors and sent to the system that needs to analyze or respond in time. Instead of storing the entire stream, continuous online learning faces one input u_t at a time.

The rigorous model of a complex system is always a set of differential equations with unknown physics parameters [36]. Without losing generality, we consider a general parametrized physical system of the following form:

$$u_t = N(1, u, u^2, \dots, u_x, uu_x, \dots, u_{xx}, \dots), \quad (1)$$

where the terms with subscript x represent partial differentiation of u in space domain, “1” denotes the constant, and $N(\bullet)$ is an unknown combination of the nonlinear functions, partial derivatives, constant, and additional terms of $u(x, t)$. Our goal is to select the correct terms that are most relevant to dynamic information on real-time circumstance. In view of measurements of all considered state variables U , the right-hand side

of Eq. (1) can be expressed by multiplying the library function matrix $\Theta(U)$ with the coefficient matrix Ξ as follows:

$$U_t = \Theta(U)\Xi, \tag{2}$$

where

$$\Theta(U) = [\theta_1(U), \theta_2(U), \dots, \theta_p(U)]. \tag{3}$$

Columns of $\Theta(U)$, $\theta_s(U)$ ($s = 1, 2, \dots, p$), correspond to p specific candidate terms for the governing equation, as shown in Fig. 1(b).

O-SINDy is divided into two steps: (i) build complete libraries of candidate terms; (ii) update the structure of governing equations via the FTRL-proximal style methodology. Each procedure is described as follows.

A. Build a candidate library

We start by collecting the spatiotemporal series data at m time points and n spatial locations of the state variables. For each state variable, the captured state measurements are represented as a single column state vector $u \in C^{nm \times 1}$. Based on all the observables, e.g., variables $x, y,$ and z in Lorenz systems, a series of functional terms associated with these quantities can be calculated and then reshaped into a single column as well, such as $x^2, xy, xyz, x^2y,$ etc. Likewise, partial differential terms should be considered in the candidate library if partial differential equations (PDEs) govern the dynamics. Furthermore, “1” denotes the constant term that possibly appears in equations. The compositive function library $\Theta(U) \in C^{nm \times p}$ is a matrix that contains p designed functional terms. Note that the computed time derivative $u_t \in C^{nm \times 1}$ is also a single-column vector presented on the left-hand side of Eq. (2). The constant term in the library Θ sufficiently considers the bias term in the governing differential equation so that the model can be regarded as an unbiased representation of dynamics. For O-SINDy, an important prerequisite for revealing the correct governing differential equation is that the candidate function library contains all the members which constitute the concise dynamics expression, so as to ensure that the exact sparse coefficient Ξ is computed under iterations.

B. Optimization

Formulating a hypothesis that the governing differential equations are evolving at any moment, we model each example as a dynamic process to simulate the arrival of streaming data. Given a set of time-series state measurements at a fixed number of spatial locations in x , the goal is to construct the form of $N(\bullet)$ online. To fix this issue, the core of our innovation is utilizing real-time streaming data to denote the loss function. Considering a coefficient vector in Ξ , ξ_j ($j = 1, 2, \dots, d$), which corresponds to the specific state variable u , the fitness function is designed as follows:

$$\min_{\xi_j} = \sum_{i=1}^{mn} L_i(\xi_j) + \lambda_1 \|\xi_j\|_1, \tag{4}$$

where $\lambda_1 \geq 0$ denotes the regularization coefficient, n is the number of positions, m denotes the length of the time series,

$\xi_j = [\xi_{j1}, \xi_{j2}, \dots, \xi_{jp}]^T$, and $L_i(\xi_j)$ is defined as follows:

$$L_i(\xi_j) = \frac{(\Theta_i \xi_j - \dot{U}_i)^2}{2}, \tag{5}$$

where \dot{U}_i is the time derivative of the i th state observation and Θ_i is the i th row of the common library Θ that represents all candidate function values for the i th data. The sparsity constraints mean that the coefficient vector ξ_j is sparse with only a few nonzero entries, each representing a subsistent item in the function library. Subsequently, we exploit the follow-the-regularized-leader (FTRL)-proximal [37–39] style methodology to optimize the outcome by considering solely one instance each time.

Leveraging the fact that each instance is considered individually, we rewrite the loss function and define a single loss term [see Eq. (5)]. For example, if the first state measurement u_1 is available, the loss is defined as $L_1(\xi_j) + \lambda_1 \|\xi_j\|_1 = 0.5(\Theta_1 \xi_j - \dot{U}_1)^2 + \lambda_1 \|\xi_j\|_1$. Next, if the second instance of u_2 arrives, the loss is defined as $L_1(\xi_j) + L_2(\xi_j) + \lambda_1 \|\xi_j\|_1 = 0.5(\Theta_1 \xi_j - \dot{U}_1)^2 + 0.5(\Theta_2 \xi_j - \dot{U}_2)^2 + \lambda_1 \|\xi_j\|_1$. In this way, the computer only needs to store information about a single example, thus relieving memory stress. On the i th sample, the gradient is calculated as follows:

$$\nabla L_i(\xi_j) = (\Theta_i \xi_j - \dot{U}_i) \Theta_i. \tag{6}$$

Normally, the online gradient descent algorithm can be used to update the coefficient vector ξ_j^i after the arrival of the i th data by using

$$\xi_j^{i+1} = \xi_j^i + C \nabla L_i(\xi_j). \tag{7}$$

However, this method has been proven to lack general applicability [38]. Correspondingly, the FTRL-proximal style approach is introduced to solve the online problem. We use the following equation to update the coefficient vector ξ_j^i :

$$\begin{aligned} \xi_j^{i+1} = \arg \min_{\xi_j} & \sum_{k=1}^i \nabla L_k(\xi_j^k) \xi_j + \lambda_1 \|\xi_j\|_1 \\ & + 0.5 \lambda_2 \|\xi_j\|_2^2 + 0.5 \sum_{k=1}^i \sigma_k \|\xi_j - \xi_j^k\|_2^2, \end{aligned} \tag{8}$$

where λ_1 and λ_2 both denote the regularization coefficient, which is the positive constant, while σ_k is on the i th learning rate η_i aspect, defined by

$$\sum_{k=1}^i \sigma_k = \sqrt{t} = \eta_i^{-1}. \tag{9}$$

Giving the gradient $\nabla L_i(\xi_j)$ a shorthand g_i , we set $g_{1:i} = \sum_{k=1}^i g_k$. Then the update in Eq. (8) can be rewritten as follows:

$$\begin{aligned} \xi_j^{i+1} = \arg \min_{\xi_j} & \left(g_{1:i} - \sum_{k=1}^i \sigma_k \xi_j^k \right) \xi_j + \lambda_1 \|\xi_j\|_1 \\ & + 0.5 \left(\lambda_2 + \sum_{k=1}^i \sigma_k \right) \|\xi_j\|_2^2 + 0.5 \sum_{k=1}^i \sigma_k \|\xi_j^k\|_2^2, \end{aligned} \tag{10}$$

where the last term $0.5 \sum_{k=1}^i \sigma_k \|\xi_j^k\|_2^2$ is a constant with respect to ξ_j and negligibly impacts the update process. Let $Z_i = g_{1:i} - \sum_{k=1}^i \sigma_k \xi_j^k = Z_{i-1} + g_i - \sigma_i \xi_j^i$, and we ignore the last term and rewrite Eq. (10) as

$$\arg \min_{\xi_j} \left[Z_i \xi_j + \lambda_1 \|\xi_j\|_1 + 0.5 \left(\lambda_2 + \sum_{k=1}^i \sigma_k \right) \|\xi_j\|_2^2 \right]. \quad (11)$$

Let ξ_{js} and $Z_{i,s}$ ($s = 1, 2, \dots, p$) represent the s th element of the vector ξ_j and Z_i , respectively. Equation (9) is generalized as

$$\begin{aligned} \xi_{js}^{i+1} = \arg \min_{\xi_{js}} & Z_{i,s} \xi_{js} + \lambda_1 |\xi_{js}| \\ & + 0.5 \left(\lambda_2 + \sum_{k=1}^i \sigma_k \right) \xi_{js}^2. \end{aligned} \quad (12)$$

Aiming to simplify the expression, we suppose w^* to be the optimal solution of ξ_{js}^i and $\Phi \in \nabla w^*$ to be its gradient. Then, Eq. (13) is satisfied. Equation (14) shows the subdifferential of $w = \text{sgn}(w)$:

$$\begin{aligned} Z_{i,s} + \lambda_1 \Phi + \left(\lambda_2 + \sum_{k=1}^i \sigma_k \right) w^* &= 0, \quad (13) \\ \text{sgn}(w) = \begin{cases} \Phi \in R | -1 \leq \Phi \leq 1, & \text{if } w = 0, \\ 1, & \text{if } w > 0, \\ -1, & \text{if } w < 0. \end{cases} \end{aligned} \quad (14)$$

The solution to w^* can be discussed in the following three cases [40].

- (1) If $|Z_{i,s}| \leq \lambda_1$,
- (a) If $w^* = 0$, then $\text{sgn}(0) \in (-1, 1)$ and Eq. (13) is satisfied.
- (b) If $w^* > 0$, then $Z_{i,s} + \lambda_1 \text{sgn}(w^*) = Z_{i,s} + \lambda_1 \geq 0$ and $(\lambda_2 + \sum_{k=1}^i \sigma_k) w^* > 0$. Equation (13) does not hold.
- (c) If $w^* < 0$, then $Z_{i,s} + \lambda_1 \text{sgn}(w^*) = Z_{i,s} - \lambda_1 \leq 0$ and $(\lambda_2 + \sum_{k=1}^i \sigma_k) w^* < 0$. Equation (13) does not hold.
- (2) If $|Z_{i,s}| < -\lambda_1$,
- (a) If $w^* = 0$, then $\text{sgn}(0) \in (-1, 1)$ and Eq. (13) does not hold.
- (b) If $w^* > 0$, then $Z_{i,s} + \lambda_1 \text{sgn}(w^*) = Z_{i,s} + \lambda_1 < 0$ and $(\lambda_2 + \sum_{k=1}^i \sigma_k) w^* > 0$. Equation (13) holds and the solution is

$$w^* = - \left(\lambda_2 + \sum_{k=1}^i \sigma_k \right)^{-1} (Z_{i,s} + \lambda_1). \quad (15)$$

- (c) If $w^* < 0$, then $Z_{i,s} + \lambda_1 \text{sgn}(w^*) = Z_{i,s} - \lambda_1 < 0$ and $(\lambda_2 + \sum_{k=1}^i \sigma_k) w^* < 0$. Equation (13) does not hold.
- (3) If $|Z_{i,s}| > \lambda_1$,
- (a) If $w^* = 0$, then $\text{sgn}(0) \in (-1, 1)$ and Eq. (13) does not hold.
- (b) If $w^* > 0$, then $Z_{i,s} + \lambda_1 \text{sgn}(w^*) = Z_{i,s} + \lambda_1 > 0$ and $(\lambda_2 + \sum_{k=1}^i \sigma_k) w^* > 0$. Equation (13) does not hold.
- (c) If $w^* < 0$, then $Z_{i,s} + \lambda_1 \text{sgn}(w^*) = Z_{i,s} - \lambda_1 > 0$ and $(\lambda_2 + \sum_{k=1}^i \sigma_k) w^* < 0$. Equation (13) holds and the solution is

$$w^* = - \left(\lambda_2 + \sum_{k=1}^i \sigma_k \right)^{-1} (Z_{i,s} - \lambda_1). \quad (16)$$

Algorithm 1: O-SINDy.

```

Input:  $\alpha, \beta, \lambda_1, \lambda_2$ : the parameters of O-SINDy;
thresh: the threshold for truncation;
epo: the iterations for reusing the finite data;
 $\Theta, u_i$ : the pre-processed training data;
Output:  $\xi_j$ ;
1 Set  $N$  as the length of training data and  $p$  as the
  column number in  $\Theta$ ;
2 for each term  $s$  in 1:  $p$  do
3   Set  $Z_s = 0, \eta_s = 0$ , and  $\xi_{js} = 0$ ;
4 end
5 for each epoch in 1:  $epo$  do
6   for each instance  $i$  in 1:  $N$  do
7     Receive instance pair  $(\Theta_i, \hat{U}_i)$  and set
        $I = \{k | \Theta_{ik} \neq 0\}$ ;
8     for each element  $k$  in  $I$  do
9       Compute  $\xi_{jk}^i$  using Eq. (20);
10    for each element  $k$  in  $I$  do
11       $g = (\Theta_i \xi_j^i - \hat{U}_i) \Theta_{ik}$ ;
12       $\sigma = (\sqrt{\eta_k + g^2} - \sqrt{\eta_k}) / \alpha$ ;
13       $Z_k = Z_k + g - \sigma \xi_{jk}^i$ ;
14       $\eta_k = \eta_k + g^2$ ;
15    end
16  end
17 end
18 end
19 for each term  $s$  in 1:  $p$  do
20   Set  $\xi_{js} = 0$  if  $|\xi_{js}| < \text{thresh}$ ;
21 end

```

The above discussion in different estimation scenarios is summarized in Eq. (17), where $w^* = \xi_{js}^i$:

$$w^* = \begin{cases} 0, & |Z_{i,s}| \leq \lambda_1, \\ -\frac{[Z_{i,s} - \lambda_1 \text{sgn}(Z_{i,s})]}{(\lambda_2 + \sum_{k=1}^i \sigma_k)}, & \text{otherwise.} \end{cases} \quad (17)$$

Moreover, we set a remarkable learning rate

$$\eta_{is} = \frac{\alpha}{(\beta + \sqrt{\sum_{k=1}^i (g_{k,s})^2})}, \quad (18)$$

where α and β are both positive constants, $g_{k,s}$ denotes the s th element of the gradient g_k , and η_{is} is the learning rate of $g_{k,s}$. Thus Eq. (9) can be rewritten as follows:

$$\sum_{k=1}^i \sigma_k = \frac{1}{\eta_{is}} = \frac{(\beta + \sqrt{\sum_{k=1}^i (g_{k,s})^2})}{\alpha}. \quad (19)$$

The final form of the closed solution is obtained and expressed as

$$\xi_{js}^i = \begin{cases} 0, & |Z_{i,s}| \leq \lambda_1, \\ \frac{[\lambda_1 \text{sgn}(Z_{i,s}) - Z_{i,s}]}{(\lambda_2 + \frac{(\beta + \sqrt{\sum_{k=1}^i (g_{k,s})^2})}{\alpha})}, & \text{otherwise,} \end{cases} \quad (20)$$

where α and β are both positive constants and $g_{k,s}$ denotes the s th element of the gradient g_k .

Notably, ξ_{js} is always updated according to the current input instance, thereby guiding the detection of change points if there exists any variation in the evolving system. Additionally,

TABLE I. Summary of online identification results for a wide range of canonical models. O-SINDy is applied to reconstruct the correct model structure in each example. Settings for the spatial and temporal sampling of the numerical simulation data is given, along with the relative L_2 error in recovering the parameters of these dynamical models.

Dynamic system	Identified governing equation	Error	Setting
Harmonic oscillator 1	$\dot{x} = -0.1000x + 2.0000y$ $\dot{y} = -2.0000x - 0.1000y$	1.62×10^{-5}	$[x_0, y_0] = [2, 0]$, $m = 2500$
Harmonic oscillator 2	$\dot{x} = -0.0998x^3 + 2.0014y^3$ $\dot{y} = -2.0011x^3 - 0.1001y^3$	6.28×10^{-4}	$[x_0, y_0] = [2, 0]$, $m = 2500$
Lorenz system 1	$\dot{x} = -9.9990x - 10.8988y$ $\dot{y} = -0.8999y + 9.6000xz$ $\dot{z} = -7.1000 - 2.6998z - 3.1000xy$	2.80×10^{-4}	$[x_0, y_0, z_0] = [-8, 7, 27]$, $m = 10000$
Lorenz system 2	$\dot{x} = 5.7004z - 3.5007xy + 2.1001yz$ $\dot{y} = -10.3001 - 2.6998y + 2.5999x^2 - z^2$ $\dot{z} = -10.8828z - 5.5957xy + 3.3992yz$	2.70×10^{-4}	$[x_0, y_0, z_0] = [-8, 7, 27]$, $m = 10000$
Hopf normal form	$\dot{x} = 0.2514x - 0.9995y - 1.0060x^3 - 1.0009xy^2$ $\dot{y} = 1.0006x + 0.2606y - 1.0422x^2y - 1.0392y^3$	1.85×10^{-2}	$[x_0, y_0] = [2, 0]$, $m = 3000, \mu = 0.25, \omega = 1, A = 1$
Diffusion from random walk	$u_t = 0.5000u_{xx}$	1.69×10^{-5}	$t \in [0, 0.02], m = 3, n = 8000$
Burgers'	$u_t = -0.9993uu_x + 0.1002u_{xx}$	6.97×10^{-4}	$t \in [0, 10], m = 101$ $x \in [-8, 8], n = 256$
Korteweg–de Vries (KdV)	$u_t = -6.1317uu_x - 1.0029u_{xxx}$	2.17×10^{-2}	$t \in [0.05, 0.175], m = 6,$ $x \in [-10, 12], n = 256$
Kuramoto-Sivashinsky	$u_t = -0.9667uu_x - 0.9585u_{xx} - 0.9646u_{xxx}$	2.55×10^{-2}	$t \in [0, 100], m = 251,$ $x \in [0, 100], n = 1024$
Reaction-diffusion	$u_t = 0.1000u_{xx} + 0.1001u_{yy} - 1.0001uv^2 -$ $1.0001u^3 + 0.9996v^3 + 0.9996u^2v + 1.0001u$ $v_t = 0.1001v_{xx} + 0.1001v_{yy} + 1.0001v -$ $0.9996uv^2 - 0.9996u^3 - 1.0001v^3 - 1.0001u^2v$	2.65×10^{-4}	$t \in [0, 10], m = 201,$ <i>subsample</i> 0.285, $x, y \in [-10, 10], n = 512$

the mechanism frees the computer from storing the whole large-scale data. It should be highlighted that the method suggested above is entirely different from the batch learning methods, which update gradients depending on all available measurements. O-SINDy [41] takes advantage of a single available instance and updates gradients based on the current loss $L_i(\xi_j)$ ($i = 1, 2, \dots, mn$) in each iteration, thereby being easily extended to handle evolving systems. Note that the optimum selection of hyperparameters in each case is determined by grid search (10^i , $i = -8, -7, \dots, 1$).

III. RESULTS

A. Discovering single system from streaming data

1. Example: Discovering chaotic Lorenz system

The online algorithmic procedure for identifying correct governing differential equations of the chaotic Lorenz system from time series is demonstrated in Fig. 1. Our proposal combines data collection, a library of potential candidate functions, and the online reconstruction method. The Lorenz system is highly nonlinear described by the following equations, according to which we construct a canonical dynamic instance for model discovery:

$$\dot{x} = \sigma(y - x), \quad (21)$$

$$\dot{y} = x(\rho - z) - y, \quad (22)$$

$$\dot{z} = xy - \beta z. \quad (23)$$

To indicate the nonlinear combination of Lorenz variables, common parameters are $\sigma = 10$, $\beta = 8/3$, and $\rho = 28$, with the initial condition of $(x_0, y_0, z_0)^T = (-8, 7, 27)^T$. Reference [3] has employed the SINDy autoencoder to discover a parsimonious model with only seven active terms that seems to mismatch the original Lorenz system. Interestingly, the dynamics of the resulting model exhibit an attractor with a two-lobe structure, which is qualitatively consistent with the true Lorenz attractor. Then the sparsity pattern can be rewritten in the standard form by choosing an appropriate variable transformation. Considering the chaotic nature, O-SINDy show the ability to capture the concise model on two Lorenz systems. Results in Fig. 1(d) and Table I imply that O-SINDy can accurately reproduce the attractor dynamics from chaotic trajectory measurements. In both cases, we conducted 10 000 time steps with time intervals $dt = 0.01$.

Given time intervals and initial conditions, we simulate the Lorenz system within a certain time horizon. Specifically, state measurements are collected under the constraints of the transformed expressions, including historical data of the state U and its time derivative U_t . The combination of spatial and temporal modes results in the trajectory of dynamics shown in Fig. 1(a). A library of potential candidate functions, $\Theta(U)$, is constructed to find the least terms required to satisfy $U_t = \Theta(U)\Xi$. Candidate functions can be polynomial functions, trigonometric functions, exponential functions, logarithmic functions, partial derivatives, constant items, and any additional terms about U . The proposed online technique

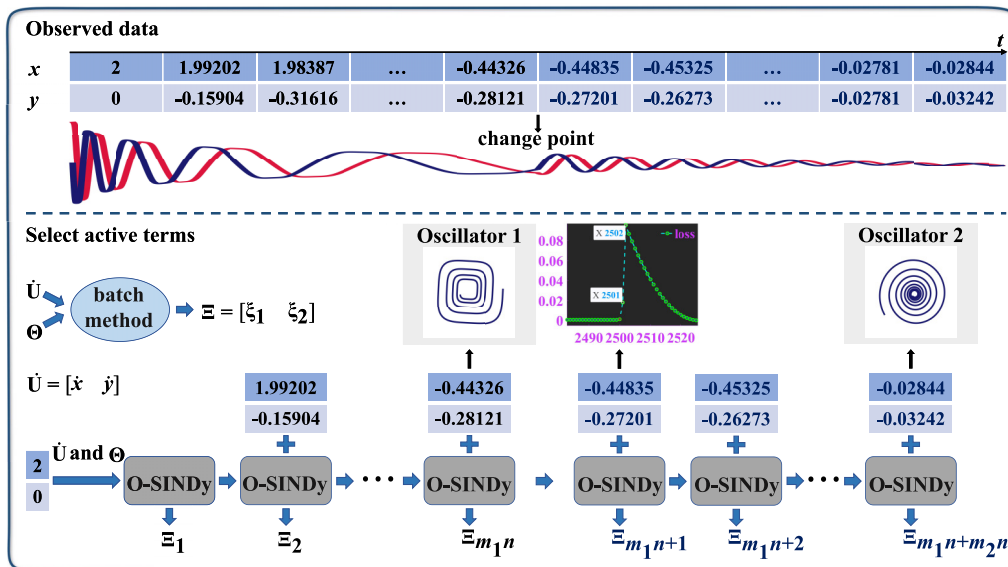


FIG. 2. Difference between batch learning methods and O-SINDy in reconstructing governing equations from evolving systems. As instances successively arrive, the change point arises when the system is disturbed and the original distribution of measurements varies. The batch method can only strike a compromise solution utilizing all data generated by the two completely different systems. By means of O-SINDy, the loss value gradually decreases and stabilizes unless the change point appears. The former governing equation is reconstructed right before the change point and the latter is identified after a few iterations.

can identify the coefficient $\Xi = [\xi_1; \xi_2; \dots; \xi_d]$ to determine active terms of the dynamics. To elude the obstacle of large-scale data storage and batch gradient update, we design a cumulative loss function to reflect the mode of streaming data by solely analyzing the information of one sample at a time. The preestablished sparse scenario guarantees that the resulting solution obtained by the online method can effectively balance model complexity with description ability to avoid overfitting, thereby promoting its interpretability and extensibility. The prompt update of components and coefficients in the governing equation is one implementation of data arrival based synchronization. Consequently, we can subtly detect moderate or drastic variations in the system through changes in the resulting model.

2. Performance on discovery for canonical models

We perform O-SINDy on ten canonical models in mathematical physics and engineering sciences and the results are shown in Table I. These physical systems contain dynamics with periodic to chaotic behavior, ranging from linear to strongly nonlinear systems. The settings for spatial and temporal sampling and the coefficient errors are detailed in Table I. Encouragingly, O-SINDy can recover every physical system, even those with significant spatial subsampling. The notable results highlight the broad applicability of the method and the success of this online technique in discovering governing differential equations. Remarkably, the capacity for capturing nontrivial active terms, in particular, has essential explanatory implications for model discovery.

B. Discovering hybrid systems

We first consider an evolving two-dimensional damped harmonic oscillator that changes from cubic to linear:

$$\frac{d \begin{bmatrix} x \\ y \end{bmatrix}}{dt} = \begin{bmatrix} -0.1 & 2 \\ -2 & -0.1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{24}$$

$$\frac{d \begin{bmatrix} x \\ y \end{bmatrix}}{dt} = \begin{bmatrix} -0.1 & 2 \\ -2 & -0.1 \end{bmatrix} \begin{bmatrix} x^3 \\ y^3 \end{bmatrix}. \tag{25}$$

A particle in simple harmonic vibration is known as a harmonic oscillator, whose motion is the simplest ideal vibration model. Figure 2 illustrates the dynamic data and the detection of change points. First, we create the solution to Eq. (25), U_1 , with 2500 time steps and the initial data [2,0]. Then, we utilize the last instance in U_1 as the initial data and create the solution to Eq. (24) with 2500 time steps as well, recorded as U_2 . Specifically, the change point arises when the system changes and then the original distribution of generated data varies. Therefore, the corresponding loss value gradually decreases and stabilizes unless the change point appears. In this context, the coefficients in the former governing equation and the change location are simultaneously identified according to the sudden increase of loss value.

With an augmented nonlinear library including polynomials up to the fifth order, the change point is spotted in evolving circumstances. The loss curve tends to stabilize faster after resetting the training coefficient, thereby accelerating the convergence of O-SINDy because the former result may mislead the training process.

The transition from the KdV to the Kuramoto-Sivashinsky equation is simulated as an illustrative example that exhibits qualitatively different dynamics as the system evolves.

TABLE II. Summary of O-SINDy and standard machine learning methods for identifying the evolving systems. O-SINDy is successfully applied to reconstruct the correct governing PDEs before and after the system changes, while the standard machine learning methods strike a compromise solution utilizing all samples.

Methodology	Cubic_Linear	KdV_KS
O-SINDy	<p>Phase 1</p> $\dot{x} = -0.1257x^3 + 2.0006y^3$ $\dot{y} = -2.0193x^3 - 0.1179y^3$ <p>Phase 2</p> $\dot{x} = -0.1006x + 1.9958y$ $\dot{y} = -1.9999x - 0.1000y$	<p>Phase 1</p> $u_t = -6.1036uu_x - 0.9460u_{xxx}$ <p>Phase 2</p> $u_t = -0.9894uu_x - 0.9713u_{xx} - 0.9158u_{xxx}$
STRidge [4]	$\dot{x} = 0.5366y + 1.3167y^3 - 0.4861x^2y + 0.1735y^5 + 0.1678x^2y^3 + 0.1170x^4y$ $\dot{y} = -0.5751x + 0.5912xy^2 + 0.1335x^2y - 1.3422x^3 - 0.1340xy^4 - 0.2122x^3y^2 - 0.1503x^5$	$u_t = -1.6007u_x + 1.4328uu_{xx} - 0.5409u^2u_{xx} - 0.5814u_{xxx} + 0.8952uu_{xxx} + 0.5741uu_{xxxx}$
TrainSTRidge [4]	$\dot{x} = 0.5340y + 1.3186y^3 - 0.4700x^2y + 0.1734y^5 + 0.1610x^2y^3 + 0.1114x^4y$ $\dot{y} = -0.5751x + 0.5912xy^2 + 0.1335x^2y - 1.3422x^3 - 0.1340xy^4 - 0.2122x^3y^2 - 0.1503x^5$	$u_t = -1.6007u_x + 1.4328uu_{xx} - 0.5409u^2u_{xx} - 0.5814u_{xxx} + 0.8952uu_{xxx} + 0.5741uu_{xxxx}$
STLS [42]	$\dot{x} = 0.4920y + 1.4390y^3 - 0.4630x^2y + 0.1310y^5 + 0.1120x^2y^3 + 0.1320x^4y$ $\dot{y} = -0.5260x + 0.4480xy^2 - 0.1320x^2y - 1.4280x^3 - 0.0830xy^4 - 0.1510x^3y^2 - 0.1240x^5$	$u_t = -1.6007u_x + 1.4328uu_{xx} - 0.5409u^2u_{xx} - 0.5814u_{xxx} + 0.8952uu_{xxx} + 0.5741uu_{xxxx}$
Lasso [21]	$\dot{x} = 0.5367y + 1.3166y^3 - 0.4872x^2y + 0.1735y^5 + 0.1681x^2y^3 + 0.1179x^4y$ $\dot{y} = -0.5748x + 0.5925xy^2 + 0.1340x^2y - 1.3340x^3 - 0.1339xy^4 - 0.2136x^3y^2 - 0.1495x^5$	$u_t = -1.6007u_x + 1.4328uu_{xx} - 0.5409u^2u_{xx} - 0.5814u_{xxx} + 0.8952uu_{xxx} + 0.5741uu_{xxxx}$
ElasticNet [43]	$\dot{x} = 0.4364y + 1.3402y^3 + 0.1751y^5$ $\dot{y} = -0.4410x - 0.0945y^3 - 1.4026x^3 - 0.1417x^5$	$u_t = -1.6007u_x + 1.4328uu_{xx} - 0.5409u^2u_{xx} - 0.5814u_{xxx} + 0.8952uu_{xxx} + 0.5741uu_{xxxx}$
FoBaGreedy [44]	$\dot{x} = 0.2302y + 1.8805y^3 - 0.1003x^3$ $\dot{y} = -0.2487x - 0.0988y^3 - 1.8757x^3$	$u_t = -1.5771u_x + 1.5007uu_{xx} - 0.5332u^2u_{xx} - 0.6168u_{xxx} + 0.9653uu_{xxx}$

Summary of O-SINDy and the state-of-the-art methods in SINDy for this identification in Table II demonstrates that our approach works well. Most data-driven methods regard batch measurements as a whole and update gradients integrally to obtain an ephemeral fitting. Owing to the ignorance of variations, existing methodologies are biased to strike a compromise solution, so that strenuous to identify the changes from consecutively generated data. Quite the contrary, O-SINDy succeeds in simultaneously estimating forms and identifying changes of the governing equations.

Moreover, we investigate the susceptible-infected-recovered (SIR) disease model with time-varying transmission rates, widely studied in epidemiology [45,46]. The following equation is a mathematical description of the SIR model:

$$\dot{S} = vN - \frac{\beta(t)}{N}IS - dS, \tag{26}$$

$$\dot{I} = \frac{\beta(t)}{N}IS - (\gamma + d)I, \tag{27}$$

$$\dot{R} = \gamma I - dR, \tag{28}$$

$$\beta(t) = \begin{cases} \hat{\beta}(1 + b), & t \in \text{Spring/Fall}, \\ \hat{\beta}\frac{1}{1+b}, & t \in \text{Summer/Winter}. \end{cases} \tag{29}$$

It is a time-dependent hybrid dynamical system, where $v = d = 0.0027$, $N = 1000$ is the total population, $\gamma = 0.2$, $b =$

0.8 , and $\hat{\beta} = 9.336$ is a base transmission rate. We simulate the model for three years within the initial condition at $[S_0, I_0, R_0] = [12, 13, 975]$, recording at a daily interval. A random perturbation is added to the start of each season with the same probability.

To illustrate the dynamics of the SIR model, we display the recovered population by subtracting 920 and scaling the

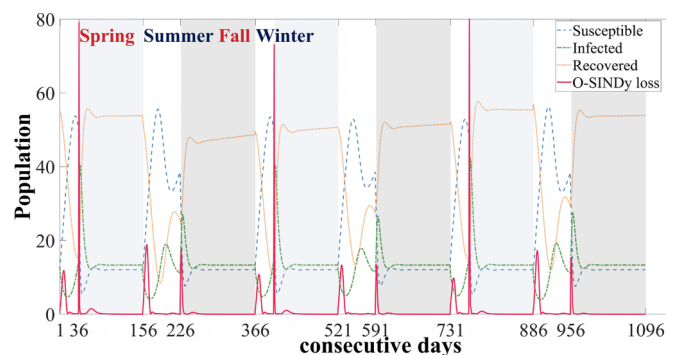


FIG. 3. Results of O-SINDy in the SIR disease model with varying transmission rates. The infected population dynamics over 3 years show declines in summer and winter, followed by outbreaks in spring and fall. The loss peak in O-SINDy appears with the varying $\beta(t)$.

TABLE III. Results of O-SINDy and Hybrid-SINDy in identifying the SIR model of four random days. O-SINDy provides better reconstructions for a single day, while Hybrid-SINDy misses some significant terms.

Date time	True models	O-SINDy	Hybrid-SINDy
30 (Spring)	$\dot{S} = 2.7397 - 0.0052IS - 0.0027S$ $\dot{I} = 0.0052IS - 0.2027I$	$\dot{S} = 2.7295 - 0.0052IS - 0.0024S$ $\dot{I} = 0.0052IS - 0.2028I$	$\dot{S} = 2.7397 - 0.0052IS - 0.0027S$ $\dot{I} = 0.0730I$
80 (Summer)	$\dot{S} = 2.7397 - 0.0168IS - 0.0027S$ $\dot{I} = 0.0168IS - 0.2027I$	$\dot{S} = 2.7397 - 0.0168IS - 0.0027S$ $\dot{I} = 0.0168IS - 0.2027I$	$\dot{S} = 2.7243 - 0.0169IS$ $\dot{I} = 0.0168IS - 0.2027I$
184 (Fall)	$\dot{S} = 2.7397 - 0.0052IS - 0.0027S$ $\dot{I} = 0.0052IS - 0.2027I$	$\dot{S} = 2.7064 - 0.0052IS + 0.0040I$ $\dot{I} = -0.0038 + 0.0052IS - 0.2023I$	$\dot{S} = 2.7397 - 0.0052IS - 0.0027S$ $\dot{I} = 0.0777I$
350 (Winter)	$\dot{S} = 2.7397 - 0.0168IS - 0.0027S$ $\dot{I} = 0.0168IS - 0.2027I$	$\dot{S} = 2.7397 - 0.0168IS - 0.0027S$ $\dot{I} = 0.0168IS - 0.2027I$	$\dot{S} = 0.2249S + 0.2054I - 0.0339IS$ $\dot{I} = 0.0168IS - 0.2027I$

loss in O-SINDy by the maximum value. As shown in Fig. 3, O-SINDy captures the transition between states by the dramatically increasing loss. In this example, the perturbation is able to be identified utilizing solely the information of the recovered population, which is ignored in hybrid-sparse identification of nonlinear dynamics (Hybrid-SINDy) [46]. It is an offline batch learning method and outputs the governing

equation of a single day by repeatedly clustering the whole time-series data during the model selection step. The identified governing equations of four random days in the four seasons are summarized in Table III. The results show that O-SINDy provides more precise reconstruction accuracy for a single day. Specifically, instances of the same coefficients share the hyperparameters in O-SINDy. At the same time,

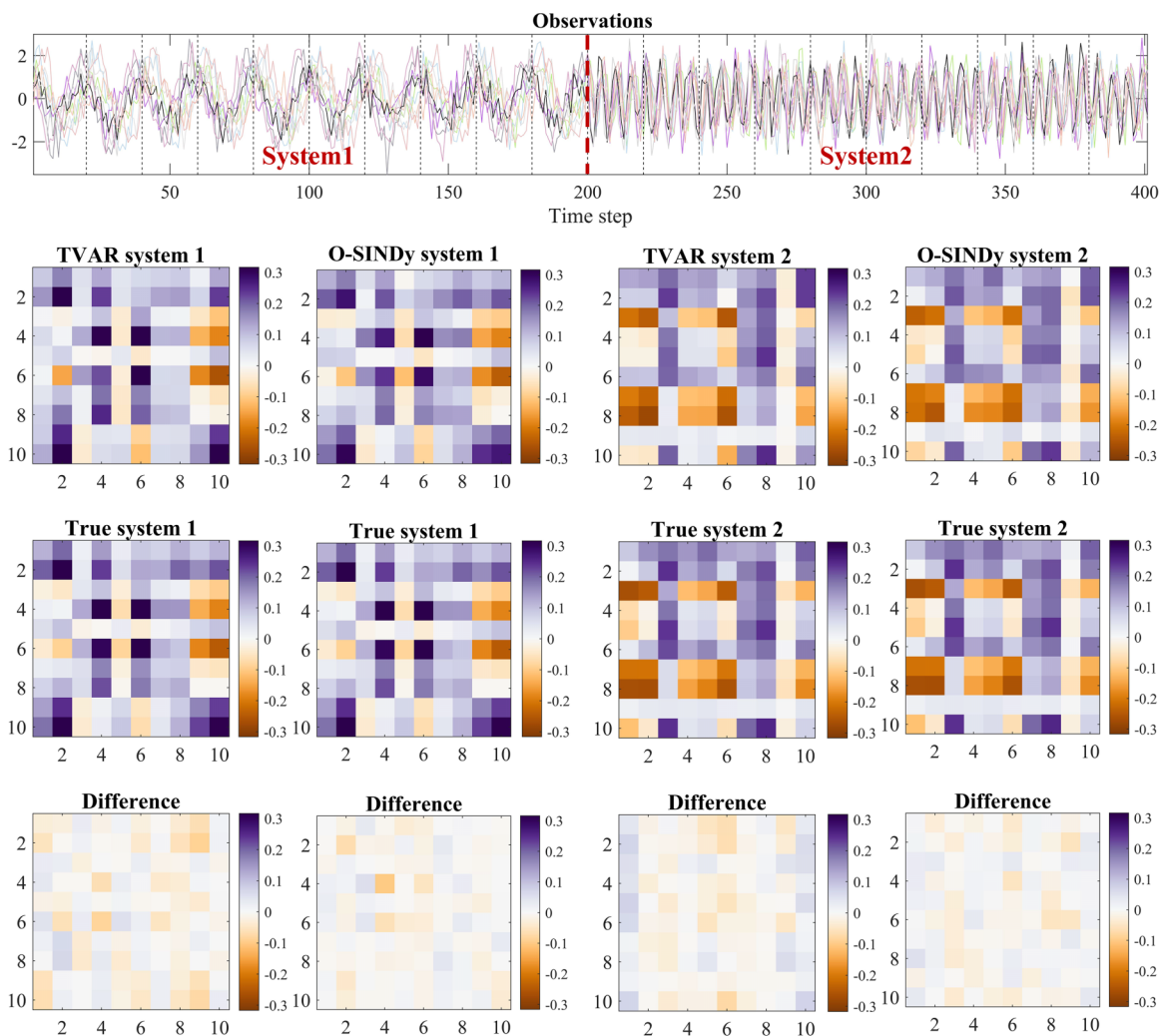


FIG. 4. Results of O-SINDy and TVART in synthetic switching linear systems. The difference between identified and true system matrices is slight in O-SINDy.

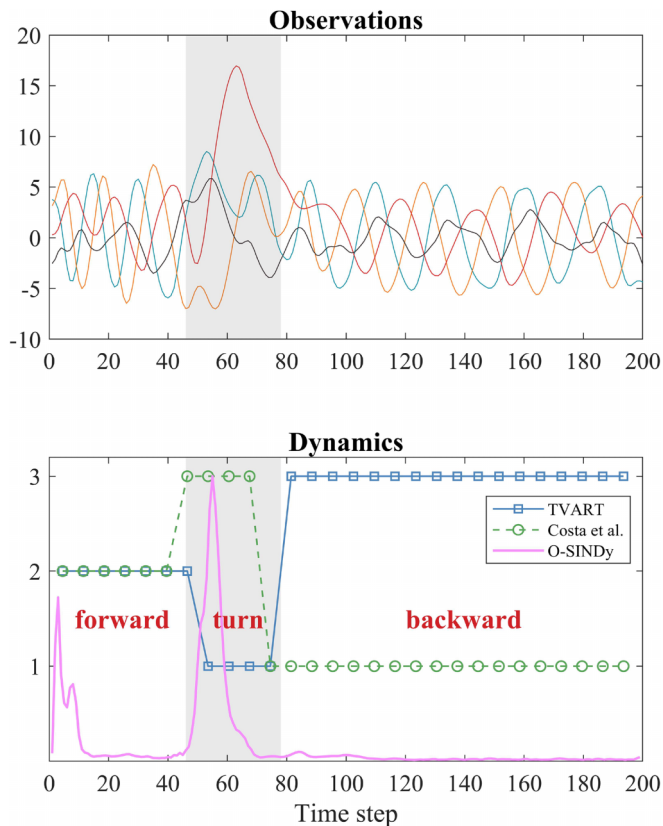


FIG. 5. Worm's posture dynamics of O-SINDy and compared methods.

Hybrid-SINDy selects a model of the highest frequency from results recovered by SINDy with different hyperparameters for each certain instance. Moreover, the effect of cluster size is particularly vital to solutions in Hybrid-SINDy.

C. Discovering switching linear systems

We also test O-SINDy on two switching linear systems, including synthetic and real-world data sets. We define $x_t \in \mathbb{R}^N$ as the sampled trajectory of a dynamical system; N is the feature dimension. The time-varying autoregressive model with low-rank tensors (TVART) [47] holds the assumption, $x_{t+1} = A_t x_t$, where A_t is constant within a time window but varies in the switching system. On the contrary, O-SINDy discards the time window and admits that A_t may change at every time point.

Regarding the synthetic example, we randomly generate two system matrices $A_{N \times N}^1$ and $A_{N \times N}^2$ to encourage the switching system. The dynamics follow $A_{N \times N}^1$ and switch to $A_{N \times N}^2$ after half of the time series. Results illustrated in Fig. 4 show that O-SINDy is competitive with the state-of-the-art technique TVART.

The posture dynamics of the worm *Caenorhabditis elegans* [48,49] is studied as the real-world example for clustering via adaptive, locally linear models [50]. This time-series data details the escape behavior of the worm in response to a heat stimulus, e.g., the transition from forward, turn, to backward crawling. Figure 5 characterizes the true dynamics and the

TABLE IV. Summary of O-SINDy for identifying the KdV equations of different spatial and temporal sampling of the numerical simulation data.

Discretization	$m = 3$ $n = 256$	$m = 6$ $n = 256$	$m = 10$ $n = 256$	$m = 10$ $n = 128$	$m = 6$ $n = 128$
Error	0.2807	0.0217	0.0090	0.0972	0.0068

discovered modes corresponding to its three typical dynamical regimes. It should be highlighted that the bifurcation of the worm dynamics is interpretable on a smaller time scale with the application of O-SINDy.

IV. CONCLUSION AND DISCUSSION

We have proposed an online modeling method, O-SINDy, capable of finding governing differential equations of evolving systems from streaming data. We demonstrate that O-SINDy works on various canonical instances. Results in Table I indicate that the inference of governing equation is accurate when utilizing O-SINDy on time series from numerical simulations. Additionally, we show the excellent performance of simultaneously estimating forms and identifying changes in time-varying dynamic systems.

More sampling points or longer time series correspond to the preferable identification of the internal control structure, while the KdV equation is an exception (see Table IV). One potential viewpoint is that approximating the soliton solution introduces great uncertainty. Nevertheless, it remains an open question to estimate the required time-series length for distilling the accurate underlying governing differential equations. The implementation of existing methodologies fundamentally depends on sufficiently large data sets, even though the dynamics are only a parsimonious representation. O-SINDy, on the contrary, is demonstrated to recover the true governing equations by iteratively reusing the finite available time series. Note that the computation of time derivatives results in the main error, which is magnified by the numerical roundoff. Thus correctly estimating numerical derivatives is the most critical and challenging task for O-SINDy, especially in a noisy context.

O-SINDy is a viable tool capable of tackling streaming data from evolving systems for accomplishing the assignment of model discovery. The integration opens up an interesting research insight for real-time modeling, online analysis, and control techniques of complex dynamic systems.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62206205, in part by the Fundamental Research Funds for the Central Universities under Grant No. XJS211905, in part by the Guangdong High-level Innovation Research Institution Project under Grant No. 2021B0909050008, and in part by the Guangzhou Key Research and Development Program under Grant No. 202206030003.

APPENDIX A: DATA PREPROCESSING

For massive-scale data sets, sparse sampling can be used to reduce the data size. For identification, it should be noted that we only require a small number of spatial points and their neighbors, whose responsibility is estimating the partial derivative terms in the candidate library. That is, local information around each measurement is necessarily wanted. Distinction allowing for application to subsampled spatiotemporal sequences, to an extent, is critically important due to experimentally and computationally prohibitive implementation of collecting full-state measurements [4]. In terms of derivation, second-order finite differences [51] are devoted to the clean data from numerical simulations, while the easiest to implement and most reliable method for the noisy data is a polynomial interpolation [52]. In principle, we abandon those points close to the boundaries due to the absence of numerical derivatives. If we know any prior knowledge about the governing equation, for instance, if one of the potential terms is determined to be nonzero in advance, we can apply the additional information to the initialization phase. Moreover, truncation of the solution is a tool to maintain its sparsity and different threshold values may provide distinguishing sparsity levels of the final output. As for the artificial noise added to the state measurements of governing equations, we use white noise. The noisy data is directly trained in experiments. In terms of the reaction-diffusion equation, exceptionally, we use the singular value decomposition (SVD) [53] to denoise some noisy spatiotemporal series and the result is a low-dimensional approximation of data sets.

APPENDIX B: CANONICAL MODELS

1. Reaction diffusion

The dynamic of reaction-diffusion systems is defined by the following equations, which provide great expressiveness and freedom although they are deceptively simple:

$$u_t = 0.1\nabla^2 u + \lambda(A)u - w(A)v, \quad (\text{B1})$$

$$v_t = 0.1\nabla^2 v + w(A)u + \lambda(A)v, \quad (\text{B2})$$

where $A = u^2 + v^2$, $w(A) = -\beta A^2$, and $\lambda(A) = 1 - A^2$. Given the vast number of data points, we subsample 5000 discretized spatial points with 30 time points each.

2. Hopf norm form

In this example, the application of O-SINDy is extended to a parametrized normal form of Poincaré-Andronov-Hopf bifurcation:

$$\dot{x} = \mu x - wy - Ax(x^2 + y^2), \quad (\text{B3})$$

$$\dot{y} = wx + \mu y - Ay(x^2 + y^2). \quad (\text{B4})$$

We explored a normal form of Hopf bifurcation on two-dimensional ordinary differential equation systems. Given $w = 1$ and $A = 1$, we collected data from the noise-free system for eight various values of the parameter μ .

3. Fokker-Planck equation

A Fokker-Planck equation that reflects the connection between the diffusion equation and Brownian motion has been taken into account. Considering the simplest form of the diffusion equation where the diffusion coefficient is 0.5 and the drift term is zero, the corresponding Fokker-Planck equation is $u_t = 0.5u_{xx}$. The Brownian motion model is usually simplified as a random walk in physics, such as the random movement of molecules in liquids and gases. To simulate the process, we add a distributed random variable with variance $dt = 0.01$ to the time series and sample the movement of the random walker.

4. Burgers' equation

Here, we consider a fundamental partial differential equation in the fields of fluid mechanics, nonlinear acoustics, and aerodynamics. As a dissipative system in one-dimensional space, the general form of Burgers' equation is described by a diffusion coefficient c (also known as kinematic viscosity). The speed of the fluid at the indicated spatial coordinate x and temporal coordinate t in a thin ideal pipe can be expressed as the following equation:

$$u_t = -uu_x + cu_{xx}, \quad (\text{B5})$$

where the diffusion term u_{xx} contributes the advective form of the Burgers' equation.

5. Korteweg-de Vries equation

The Korteweg-de Vries (KdV) equation is a nonlinear, partial differential equation for a function u of two real variables, x and t , which refer to space and time, respectively. Numerically, its solutions seem to be decomposed into a collection of well-separated solitary waves that are almost unaffected in shape by passing through each other. The soliton solution is given by

$$u(x, t) = 0.5c \times \text{sech}^2[0.5\sqrt{c}(x - ct - a)], \quad (\text{B6})$$

where c stands for the phase speed, a is an arbitrary constant, and $\text{sech}(x) = 1/\cosh(x)$ is the hyperbolic secant function. This equation describes a right-moving soliton that propagates with a speed proportional to the amplitude.

The correct equation cannot be distinguished from a single propagating soliton solution, due to the fact that some studied expressions may be solutions to more than one PDE [4]. For example, both the one-way wave equation $u_t + cu_x = 0$ and the KdV equation admit the same traveling wave solution of the form $u = f(x - ct)$ if the initial data was a hyperbolic secant squared. Hence we constructed time-series data for more than a single initial amplitude to rectify the ambiguity in selecting the governing PDE, thereby enabling the unique determination.

As the circumstance under a single propagating soliton solution, we respectively constructed two solutions without noise having the traveling speed c equal to 5 or 1 on grids with 6 time steps and 256 spatial points. Ultimately, corresponding advection equations with different c were identified using a single traveling wave. We believe that two

waves with different amplitudes and speeds may solve the KdV equation.

6. Kuramoto-Sivashinsky equation

The Kuramoto-Sivashinsky (KS) equation is a fourth-order nonlinear PDE derived by Kuramoto and Sivashinsky

in the late 1970s. Specifically, it provides two dissipative terms u_{xxx} and u_{xxxx} based on Burgers' equation, where the fourth-order diffusion term accomplishes the stabilizing regularization rather than the second-order diffusion term u_{xx} , which leads to long-wavelength instabilities. By leveraging a spectral method, the numerical solution to the KS equation was created with 101 time steps and 1024 spatial points.

-
- [1] G. Sugihara and R. M. May, Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series, *Nature (London)* **344**, 734 (1990).
- [2] J. A. Tropp and A. C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, *IEEE Trans. Inf. Theory* **53**, 4655 (2007).
- [3] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, Data-driven discovery of coordinates and governing equations, *Proc. Natl. Acad. Sci. USA* **116**, 22445 (2019).
- [4] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Data-driven discovery of partial differential equations, *Sci. Adv.* **3**, e1602614 (2017).
- [5] J. P. Crutchfield and B. McNamara, Equations of motion from a data series, *Complex Syst.* **1**, 417 (1987).
- [6] S. Li, E. Kaiser, S. Laima, H. Li, S. L. Brunton, and J. N. Kutz, Discovering time-varying aerodynamics of a prototype bridge by sparse identification of nonlinear dynamical systems, *Phys. Rev. E* **100**, 022220 (2019).
- [7] P. A. K. Reinbold and R. O. Grigoriev, Data-driven discovery of partial differential equation models with latent variables, *Phys. Rev. E* **100**, 022219 (2019).
- [8] S. Maddu, B. L. Cheeseman, C. L. Müller, and I. F. Sbalzarini, Learning physically consistent differential equation models from data using group sparsity, *Phys. Rev. E* **103**, 042310 (2021).
- [9] A. Somacal, Y. Barrera, L. Boechi, M. Jonckheere, V. Lefieux, D. Picard, and E. Smucler, Uncovering differential equations from data with hidden variables, *Phys. Rev. E* **105**, 054209 (2022).
- [10] D. E. Shea, S. L. Brunton, and J. N. Kutz, SINDy-BVP: Sparse identification of nonlinear dynamics for boundary value problems, *Phys. Rev. Res.* **3**, 023255 (2021).
- [11] H. Xu and D. Zhang, Robust discovery of partial differential equations in complex situations, *Phys. Rev. Res.* **3**, 033270 (2021).
- [12] Y. Chen, Y. Luo, Q. Liu, H. Xu, and D. Zhang, Symbolic genetic algorithm for discovering open-form partial differential equations (SGA-PDE), *Phys. Rev. Res.* **4**, 023174 (2022).
- [13] H. Ye, R. J. Beamish, S. M. Glaser, S. C. Grant, C.-h. Hsieh, L. J. Richards, J. T. Schnute, and G. Sugihara, Equation-free mechanistic ecosystem forecasting using empirical dynamic modeling, *Proc. Natl. Acad. Sci. USA* **112**, E1569 (2015).
- [14] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, C. Theodoropoulos *et al.*, Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis, *Commun. Math. Sci.* **1**, 715 (2003).
- [15] J. L. Proctor, S. L. Brunton, B. W. Brunton, and J. Kutz, Exploiting sparsity and equation-free architectures in complex systems, *Eur. Phys. J.: Spec. Top.* **223**, 2665 (2014).
- [16] A. J. Roberts, *Model Emergent Dynamics in Complex Systems* (SIAM, Philadelphia, 2014), Vol. 20.
- [17] A. J. Majda, C. Franzke, and D. Crommelin, Normal forms for reduced stochastic climate models, *Proc. Natl. Acad. Sci. USA* **106**, 3649 (2009).
- [18] D. Giannakis and A. J. Majda, Nonlinear Laplacian spectral analysis for time series with intermittency and low-frequency variability, *Proc. Natl. Acad. Sci. USA* **109**, 2222 (2012).
- [19] B. C. Daniels and I. Nemenman, Automated adaptive inference of phenomenological dynamical models, *Nat. Commun.* **6**, 8133 (2015).
- [20] H. U. Voss, P. Kolodner, M. Abel, and J. Kurths, Amplitude Equations from Spatiotemporal Binary-Fluid Convection Data, *Phys. Rev. Lett.* **83**, 3422 (1999).
- [21] R. Tibshirani, Regression shrinkage and selection via the lasso, *J. R. Stat. Soc.: Ser. B (Methodol.)* **58**, 267 (1996).
- [22] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Inferring biological networks by sparse identification of nonlinear dynamics, *IEEE Trans. Mol. Biol. Multi-Scale Commun.* **2**, 52 (2016).
- [23] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* (Cambridge University Press, Cambridge, UK, 2022).
- [24] B. Lusch, J. N. Kutz, and S. L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics, *Nat. Commun.* **9**, 4950 (2018).
- [25] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Natl. Acad. Sci. USA* **113**, 3932 (2016).
- [26] Z. Long, Y. Lu, X. Ma, and B. Dong, Pde-net: Learning pdes from data, in *International Conference on Machine Learning* (PMLR, Cambridge, MA, 2018), pp. 3208–3216.
- [27] D. Hallac, Y. Park, S. Boyd, and J. Leskovec, Network inference via the time-varying graphical lasso, in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2017), pp. 205–213.
- [28] R. Ranjan, Modeling and simulation in performance optimization of big data processing frameworks, *IEEE Cloud Computing* **1**, 14 (2014).
- [29] A. Namaki, A. H. Shirazi, R. Raei, and G. Jafari, Network analysis of a financial market based on genuine correlation and threshold method, *Physica A* **390**, 3835 (2011).
- [30] A. Ahmed and E. P. Xing, Recovering time-varying networks of dependencies in social and biological studies, *Proc. Natl. Acad. Sci. USA* **106**, 11878 (2009).
- [31] S. Myers and J. Leskovec, On the convexity of latent social network inference, in *Advances in Neural Information Processing Systems*, edited by J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (Curran Associates, Inc., 2010), Vol. 23.

- [32] R. P. Monti, P. Hellyer, D. Sharp, R. Leech, C. Anagnostopoulos, and G. Montana, Estimating time-varying brain connectivity networks from functional mri time series, *NeuroImage* **103**, 427 (2014).
- [33] Y. Cai, X. Wang, G. Joós, and I. Kamwa, An online data-driven method to locate forced oscillation sources from power plants based on sparse identification of nonlinear dynamics (SINDy), *IEEE Trans. Power Syst.* **38**, 2085 (2023).
- [34] P. Conti, G. Gobat, S. Fresca, A. Manzoni, and A. Frangi, Reduced order modeling of parametrized systems through autoencoders and sindy approach: Continuation of periodic solutions, *Comput. Methods Appl. Mech. Eng.* **411**, 116072 (2023).
- [35] D. A. Messenger, E. Dall’Anese, and D. Bortz, Online weak-form sparse identification of partial differential equations, in *Mathematical and Scientific Machine Learning* (PMLR, Cambridge, MA, 2022), pp. 241–256.
- [36] H. Schaeffer, R. Cafilisch, C. D. Hauck, and S. Osher, Sparse dynamics for partial differential equations, *Proc. Natl. Acad. Sci. USA* **110**, 6634 (2013).
- [37] B. McMahan, Follow-the-regularized-leader and mirror descent: Equivalence theorems and l1 regularization, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (PMLR, Cambridge, MA, 2011), pp. 525–533.
- [38] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin *et al.*, Ad click prediction: A view from the trenches, in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM, New York, 2013), pp. 1222–1230.
- [39] K. Wu, X. Hao, J. Liu, P. Liu, and F. Shen, Online reconstruction of complex networks from streaming data, *IEEE Trans. Cybern.* **52**, 5136 (2022).
- [40] S. S. Chen, D. L. Donoho, and M. A. Saunders, Atomic decomposition by basis pursuit, *SIAM Rev.* **43**, 129 (2001).
- [41] Available code: <https://github.com/xiaoyuans/O-SINDy>.
- [42] M. Budišić, R. Mohr, and I. Mezić, Applied koopmanism, *Chaos* **22**, 047510 (2012).
- [43] H. Zou and T. Hastie, Regularization and variable selection via the elastic net, *J. R. Stat. Soc. B* **67**, 301 (2005).
- [44] T. Zhang, Adaptive forward-backward greedy algorithm for sparse learning with linear models, in *Advances in Neural Information Processing Systems*, edited by D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou (Curran Associates, Inc., 2008), Vol. 21.
- [45] B. T. Grenfell, A. Kleczkowski, S. Ellner, and B. Bolker, Measles as a case study in nonlinear forecasting and chaos, *Philos. Trans. R. Soc. London, Ser. A* **348**, 515 (1994).
- [46] N. M. Mangan, T. Askham, S. L. Brunton, J. N. Kutz, and J. L. Proctor, Model selection for hybrid dynamical systems via sparse regression, *Proc. R. Soc. A* **475**, 20180534 (2019).
- [47] K. D. Harris, A. Aravkin, R. Rao, and B. W. Brunton, Time-varying autoregression with low-rank tensors, *SIAM J. Appl. Dyn. Syst.* **20**, 2335 (2021).
- [48] G. J. Stephens, B. Johnson-Kerner, W. Bialek, and W. S. Ryu, Dimensionality and dynamics in the behavior of *C. elegans*, *PLoS Comput. Biol.* **4**, e1000028 (2008).
- [49] O. D. Broekmans, J. B. Rodgers, W. S. Ryu, and G. J. Stephens, Resolving coiled shapes reveals new reorientation behaviors in *C. elegans*, *Elife* **5**, e17227 (2016).
- [50] A. C. Costa, T. Ahamed, and G. J. Stephens, Adaptive, locally linear models of complex dynamics, *Proc. Natl. Acad. Sci. USA* **116**, 1501 (2019).
- [51] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems* (SIAM, Philadelphia, 2007).
- [52] O. Bruno and D. Hoch, Numerical differentiation of approximated functions with limited order-of-accuracy deterioration, *SIAM J. Numer. Anal.* **50**, 1581 (2012).
- [53] M. Gavish and D. L. Donoho, The optimal hard threshold for singular values is $4/\sqrt{3}$, *IEEE Trans. Inf. Theory* **60**, 5040 (2014).