DOI: https://doi.org/10.1063/5.0160900

Chaos
An Interdisciplinary Journal
of Nonlinear Science

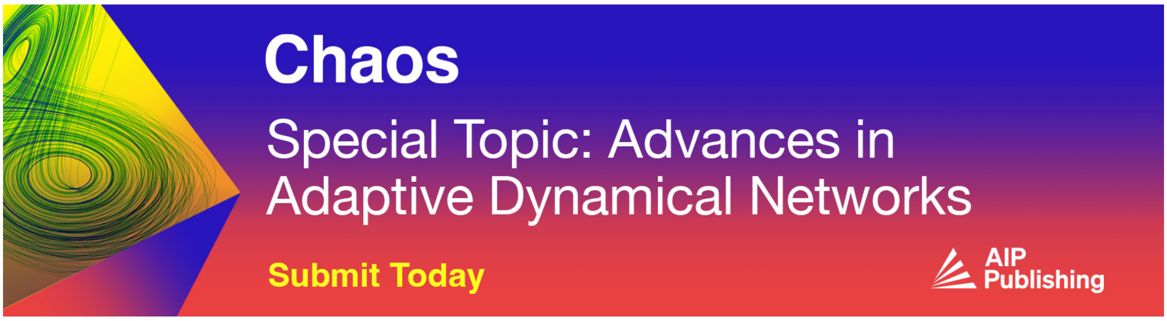
RESEARCH ARTICLE | NOVEMBER 15 2023

Machine discovery of partial differential equations from spatiotemporal data: A sparse Bayesian learning framework

Ye Yuan ; Xiuting Li; Liang Li; Frank J. Jiang ; Xiuchuan Tang; Fumin Zhang ; Jorge Goncalves ; Henning U. Voss ; Han Ding; Jürgen Kurths

Chaos 33, 113122 (2023)
https://doi.org/10.1063/5.0160900

View Online
Export Citation
CrossMark



AIP Publishing

Machine discovery of partial differential equations from spatiotemporal data: A sparse Bayesian learning framework

Cite as: Chaos **33**, 113122 (2023); doi: 10.1063/5.0160900
Submitted: 6 June 2023 · Accepted: 15 October 2023 ·
Published Online: 15 November 2023

View Online Export Citation CrossMark

Ye Yuan,[1,a)] Xiuting Li,[2] Liang Li,[3] Frank J. Jiang,[4] Xiuchuan Tang,[5] Fumin Zhang,[6] Jorge Goncalves,[7] Henning U. Voss,[8] Han Ding,[9] and Jürgen Kurths[10]

AFFILIATIONS

[1] School of Artificial Intelligence and Automation, State Key Laboratory of Digital Manufacturing Equipments and Technology, Huazhong University of Science and Technology, Wuhan 430074, People's Republic of China
[2] College of Informatics, Huazhong Agricultural University, Wuhan 430070, People's Republic of China
[3] School of Artificial Intelligence and Automation, Key Laboratory of Image Processing and Intelligent Control, Huazhong University of Science and Technology, Wuhan 430074, People's Republic of China
[4] Division of Decision and Control Systems, KTH Royal Institute of Technology, 10044 Stockholm, Sweden
[5] School of Automation, Tsinghua University, Beijing 100084, People's Republic of China
[6] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia 30309, USA
[7] Department of Engineering, University of Cambridge, Cambridge, United Kingdom and the Luxembourg Centre for Systems Biomedicine, University of Luxembourg, L-4362 Belvaux, Esch-sur-Alzette, Luxembourg
[8] Cornell MRI Facility, College of Human Ecology, Cornell University, Ithaca, New York 10065, USA
[9] School of Mechanical Science and Engineering, State Key Laboratory of Digital Manufacturing Equipments and Technology, Huazhong University of Science and Technology, Wuhan 430074, People's Republic of China
[10] Research Domain IV—Transdisciplinary Concepts & Methods, Potsdam Institute for Climate Impact Research, Potsdam D-14415, Germany

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

ABSTRACT

This study presents a general framework, namely, Sparse Spatiotemporal System Discovery (S^3d), for discovering dynamical models given by Partial Differential Equations (PDEs) from spatiotemporal data. S^3d is built on the recent development of sparse Bayesian learning, which enforces sparsity in the estimated PDEs. This approach enables a balance between model complexity and fitting error with theoretical guarantees. The proposed framework integrates Bayesian inference and a sparse *priori* distribution with the sparse regression method. It also introduces a principled iterative re-weighted algorithm to select dominant features in PDEs and solve for the sparse coefficients. We have demonstrated the discovery of the complex Ginzburg–Landau equation from a traveling-wave convection experiment, as well as several other PDEs, including the important cases of Navier–Stokes and sine-Gordon equations, from simulated data.

Data-driven methods provide a viable alternative to more traditional approaches, such as first-principles derivation, for accelerating the discovery of physical laws. This work aims to develop a sparse Bayesian learning-based framework for modeling spatiotemporal dynamical systems described by Partial Differential Equations (PDEs). The proposed framework, based on sparse Bayesian learning (SBL), can reconstruct a variety of prototypical PDEs, including the Navier–Stokes equation, the FitzHugh–Nagumo equation for nerve conduction, and the Schrödinger equation in quantum mechanics, using a small number of samples compared to state-of-the-art (SOTA) methods. In particular, we demonstrate that the data-driven discovery

method can automatically extract the first-principles-based coupled complex Ginzburg–Landau equations from data collected in a convection experiment.

I. INTRODUCTION

Recently, there has been a surge of interest in the data-driven discovery of physical laws, which applies computational methods to discover representations from data. There have been reports of many successes using data-driven methods, such as learning a nonlinear mapping from time series,[1] predicting catastrophes in nonlinear systems,[2] creating autonomous smart systems capable of self-adaption and self-regulation for short- and long-term production management,[3] equation-free modeling multi-scale/complex systems,[4] inferring models of nonlinear coupled dynamical systems,[5] reconstructing networks with stochastic dynamical processes,[6] and extracting Koopman modes from data.[7–10]

In the system identification community, a standard way is to transform the model discovery problem into a regression problem and design a good optimization algorithm to solve the problem.[11,12] There are several numerical algorithms to find solutions to problems. The first kind of such a method builds on the ordinary least-squares regression. Examples include orthogonal reduction,[13,14] the backward elimination scheme,[15,16] least-squares estimator,[17] and the alternating conditional expectation algorithm.[18–20] These approaches do have a good fitting accuracy as described by Ref. 21 which, however, requires a relatively large number of data to train. The second type of method is the sparse regression approach, where a sparse prior assumption of underlying governing equations is incorporated to tackle the outstanding issue, overfitting. By converting the discovery problem of differential equations to a sparse regression problem, Ref. 22, and later, Brunton *et al.*[23] developed a sparse identification of nonlinear dynamics framework to extract ODEs from noisy measurement data. Rudy *et al.*[24] and Schaeffer[25] consequently extended the method in Ref. 23 to recover PDEs from spatiotemporal data.

On the development of sparse optimization problem, sparse Bayesian learning (SBL) method, originally proposed in Ref. 26 and further developed in Refs. [27–29], adopts a Bayesian probabilistic framework to find sparse solutions. By treating the parameter as a random variable with sparse-inducing prior distributions determined by a set of hyperparameters and then calculating the posterior distribution of the to-be-estimated parameter given the data, it has been shown to be more general and powerful in finding a maximally sparse solution. For instance, the maximum a posteriori (MAP) estimation with a fixed weight prior is its special case;[28] the Lasso problem can be interpreted as the explicit MAP with Laplace prior.[30,31] Specifically, Wipf *et al.*[32] established the equivalence of the SBL cost function and the objectives in canonical MAP-based sparse methods (e.g., the Lasso) via a dual-space analysis.

In this work, we propose a SBL framework for data-driven PDE discovery problems. This framework integrates Bayesian inference and sparse *priori* distribution with the sparse regression method. In our sparse Bayesian method derivation, an iterative re-weighted ℓ_1-minimization algorithm is derived to approximate the original ℓ_0-minimization. Compared with PDE-FIND with the Ridge Regression, which uses a thresholding ℓ_2-minimization algorithm, ℓ_1-minimization algorithm is known to pick the sparser solution and, therefore, can better approximate the solution to the original problem. Additionally, we also analyze spatiotemporal data observed from an experiment on convection in an ethanol water mixture in a long, narrow, annular container which is heated from below and demonstrate that the proposed data-driven discovery method is able to automatically extract the governing PDEs from spatiotemporal binary-fluid convection data. This is distinct from the previous works,[24,33] in which the data used for discovering the PDEs are generated from the numerical solution to the known PDEs. Our key empirical findings are the following:

(i) We demonstrate that an interpretable first-principle model can be discovered from inherently noisy experimental data. The proposed SBL method discovers a coupled complex Ginzburg–Landau equation (CGLE) using spatiotemporal binary-fluid convection data. Furthermore, it has been proven that the discovered values of various coefficients in the CGLE, such as the group velocity, the dispersion coefficients, the saturation parameter and the coupling coefficient, can be accurately determined in terms of the theoretical values and experimental ones provided in Refs. [19,34], and 35. Moreover, the numerical solution to the discovered CGLE matches the observed experimental phenomenon.

(ii) We demonstrate that the proposed SBL framework is capable of reconstructing a variety of prototypical PDEs, including the Navier–Stokes equation in fluid mechanics,[36] the FitzHugh–Nagumo for nerve conduction,[37] the Schrödinger equation in quantum mechanics,[38] the sine-Gordon equation, and the Korteweg–de Vries equation. By comparing with the state-of-the-art PDE-FIND method in Ref. 24 and the Douglas-Rachford method in Ref. 25, the proposed algorithm is robust to measurement noise and is able to discover the underlying PDEs using a much smaller number of samples.

The rest of this paper is organized as follows: In Sec. II, a Bayesian framework for discovering PDEs from spatiotemporal data is presented. In Secs. III and IV, the proposed framework has been applied to the discovery of PDE models using both experimental and synthetic data. Conclusions and perspectives are drawn in Sec. V.

Notations: Throughout this paper, we denote vectors by lowercase letters, e.g., x, and matrices by uppercase letters, e.g., X. For a vector x and a matrix X, we denote its transpose by x^T and X^T, respectively. Furthermore, x_i refers to its ith element of x. $\|x\|_2 = \sqrt{x^T x}$ and $\|x\|_1 = \sum_\ell |x_\ell|$ denote its Euclidean norm and ℓ_1 norm, respectively. For a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, we call ∇f and $\partial^k f/\partial x_i^k$ (abbreviated as f_{x_i}) the derivative of f and the kth partial derivative of f with respect to x_i, respectively. For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, we say vector $v \in \mathbb{R}^n$ is a sub-gradient of f at $x \in \mathbb{R}^n$ if for all $y \in \mathbb{R}^n$, $f(y) \geq f(x) + v^T(y - x)$. The set of all subgradients at x is called the subdifferential at x and is denoted $\partial f(x)$.

II. S^{3}d: A BAYESIAN FRAMEWORK TO DISCOVER PDES FROM SPATIOTEMPORAL DATA

This section proposes a framework of the S^{3}d method for data-driven discovery of PDEs. The proposed S^{3}d framework is illustrated

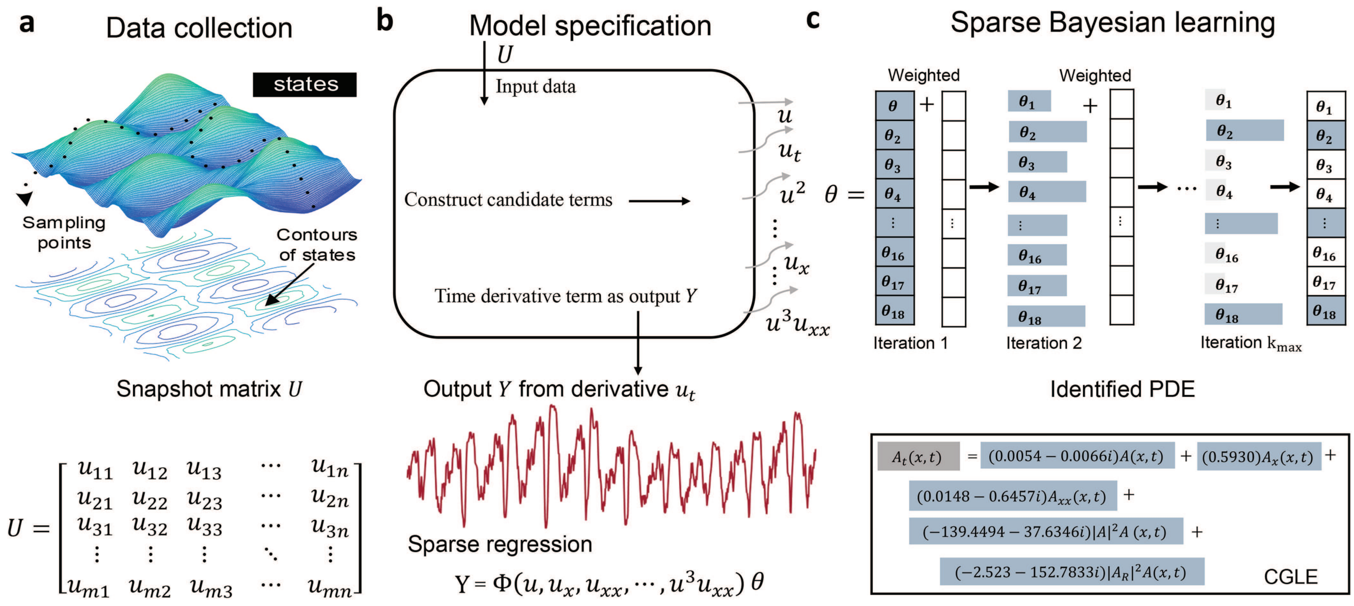

a
Data collection
states
Sampling points
Contours of states
Snapshot matrix U
$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ u_{21} & u_{22} & u_{23} & \cdots & u_{2n} \\ u_{31} & u_{32} & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & u_{m3} & \cdots & u_{mn} \end{bmatrix}$
b
Model specification
U
Input data
Construct candidate terms
Time derivative term as output Y
u
u_t
u^2
u_x
u^3u_{xx}
Output Y from derivative u_t
Sparse regression
$Y = \Phi(u, u_x, u_{xx}, \cdots, u^3u_{xx})\,\theta$
c
Sparse Bayesian learning
$\theta =$
Weighted
Weighted
θ
θ_1
θ_2
θ_3
θ_4
θ_{16}
θ_{17}
θ_{18}
Iteration 1
Iteration 2
Iteration $\mathrm{k_{max}}$
Identified PDE
$A_t(x,t) = (0.0054 - 0.0066i)A(x,t) + (0.5930)A_x(x,t) +$
$(0.0148 - 0.6457i)A_{xx}(x,t) +$
$(-139.4494 - 37.6346i)|A|^2A\,(x,t) +$
$(-2.523 - 152.7833i)|A_R|^2A(x,t)$
CGLE

FIG. 1. Schematics of the S^{3}d method used to infer the CGLE from data. (a) Data collection. Data are collected from experiments of a real system (or from a numerical solution to a PDE). The collected data are formed into a matrix. (b) Model specification. A large library of candidate terms including the potential features for the to-be-identified PDE is constructed. The candidate terms include the state u, its spatial derivatives up to two orders, and nonlinear coupling terms between them. The time derivative u_t is regarded as the model output. The collected data are used to approximate the time derivative u_t of the state u and all the candidate terms. A set of linear equations (i.e., $y = \Phi\theta$) is then defined to represent the underlying PDE. (c) Sparse Bayesian learning. Sparse regression problem is solved using an iterative reweighted ℓ_1-minimization algorithm.

in Fig. 1, which includes the following three steps: data collection, model specification, and SBL solver. We will consider the dynamics of a spatiotemporal system, which is governed by a PDE of the general form

$$u_t(x,t) = F\left(x, t, u(x,t), \frac{\partial u(x,t)}{\partial x_1}, \dots, \frac{\partial u(x,t)}{\partial x_d},\right.$$
$$\left.\times \frac{\partial^2 u(x,t)}{\partial x_1^2}, \dots, \frac{\partial^2 u(x,t)}{\partial x_1 \partial x_d}, \dots, \Theta\right), \qquad (2.1)$$

where the dynamical variable, u, is N-component with temporal and multi-dimensional spatial variables, t and $x = (x_1, \dots, x_d)^T \in \mathbb{R}^d$, respectively, defining the state. Θ is a parameter vector and F is an unknown, N-dimensional function, which depends on the dynamical variable u and its derivatives. In the following, we shall restrict the discussion to the case $d = 1$ and $N = 1$ for notational simplicity, yet the proposed framework can be applied to general cases when $d > 1$ or $N > 1$.

A. Model specification

To identify the form of F and the corresponding parameter Θ in Eq. (2.1), we construct a large library of candidate terms that may appear in the function F,

$$u_t = F\left(\frac{\partial^k u}{\partial x^k}, \frac{\partial^{k-1} u}{\partial x^{k-1}}, \dots, \frac{\partial u}{\partial x}, x; \Theta\right)$$
$$= \sum_{j=1}^{m} F_j\left(\frac{\partial^k u}{\partial x^k}, \frac{\partial^{k-1} u}{\partial x^{k-1}}, \dots, \frac{\partial u}{\partial x}, x\right)\theta_j, \qquad (2.2)$$

where $F_j(\cdot)$ is a candidate feature term, such as the polynomial, u^2, uu^2, partial derivatives, u_x, u_{xx}, u_{xxx}, or their combinations, u^2u_x, u^3u_{xx}, and $\theta = (\theta_1, \theta_2, \dots, \theta_m)^T$ is the corresponding parameter vector. A key observation we had is that θ should be sparse.[22] For instance, consider that the preselected candidate terms include the state u and the derivative of u up to the third order. Then, Eq. (2.2) can be rewritten into the following regression form:

$$u_t = \begin{bmatrix} 1 & u & u_x & u_{xx} & u_{xxx} & u^2 & \dots & u^2u_{xxx} \end{bmatrix}\theta. \qquad (2.3)$$

Given the dataset sampled on a spacetime mesh of n_x spatial points by n_t time steps, the identification problem for F is to solve the following linear equations for the parameter θ:

$$\underbrace{\begin{bmatrix} u_t(x_1,t_1) \\ u_t(x_2,t_1) \\ \vdots \\ u_t(x_{n_x},t_{n_t}) \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} 1 & u & u_x & \dots & u^2u_{xxx} \\ 1 & u & u_x & \dots & u^2u_{xxx} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & u & u_x & \dots & u^2u_{xxx} \end{bmatrix}}_{\text{Dictionary matrix } \Phi} \underbrace{\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_m \end{bmatrix}}_{\theta} \qquad (2.4)$$

where each column of Φ is a compilation of all the values of a specific candidate function on the right side of Eq. (2.3) over all grid points, and the output vector y contains the values of u_t on the left side of Eq. (2.3) over all grid points. One sequent task is to compute the time and spatial derivatives of u in Eq. (2.4). Various numerical methods including the numerical differentiation[39–41] and the symbolic or automatic differentiation[42] can be used for the estimation of the derivatives in the dictionary matrix Φ. In our framework, the polynomial approximation method is selected to estimate the derivatives from the noise-contained measurements. Indeed, the polynomial approximation method has been proved to be capable of effectively estimating the derivatives from the noise-contained data in Ref. 24. For completeness, we explain the polynomial approximation method in detail in the Appendix.

Given these approximation error, we consider the following matrix form with the noise vector ν:

$$y = \Phi\theta + \nu, \tag{2.5}$$

where the model error either arises from experimental measurements or results from numerical approximation of the derivatives included in the model output y and the dictionary matrix Φ. Thus, the type of the model error is unknown for the PDE discovery problem in practice. For simplicity, we assume that ν is i.i.d. Gaussian distributed noise with zero mean and variance σ^2, $\mathcal{N}\left(0, \sigma^2 I_n\right)$, with I_n being an $n \times n$ identity matrix.

A general solution to Eq. (2.5) can be obtained by using the least-squares method. For example, $\hat{\theta}$ is the minimizer of the following optimization problem:

$$\hat{\theta} = \arg\min_{\theta} \|y - \Phi\theta\|_2^2. \tag{2.6}$$

The solution, $\hat{\theta}$, is uniquely identified only if the number of grid points, $n = n_x n_t$, is larger than the number of columns in Φ and the matrix, Φ, has full column rank. However, for the PDE discovery problem, these two conditions, in general, are not always satisfied. Besides, if the dictionary is over-determined, the resulting optimization easily leads to overfitting.

B. Sparse Bayesian learning for S^{3}d

To tackle these issues, we take a Bayesian perspective to Eq. (2.5) by considering the likelihood of the output y is

$$p(y|\theta) = (2\pi\sigma^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}\|y - \Phi\theta\|^2\right). \tag{2.7}$$

To enforce the sparsity, we assume the following prior distribution:

$$p(\theta; \gamma) = \prod_{i=1}^{m} (2\pi\gamma_i)^{-\frac{1}{2}} \exp\left(-\frac{\theta_i^2}{2\gamma_i}\right), \tag{2.8}$$

where $\gamma := [\gamma_1, \gamma_2, \ldots, \gamma_m]^T$ is a vector of m hyperparameters. Although the hyperparametric vector γ is unknown, it determines the variance of each entry of θ and ultimately produces sparsity properties. In particular, if $\gamma_i = 0$, one can get $\mathbb{P}(\theta_i = 0) = 1$ immediately. Moreover, the hyperparametric vector can be estimated by type-Π maximum likelihood. To this end, we first compute the marginalized probability distribution

$$p(y; \gamma) = \int p(y|\theta)p(\theta; \gamma)d\theta = (2\pi)^{-\frac{n}{2}}|\Sigma_y|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}y^T\Sigma_y^{-1}y\right), \tag{2.9}$$

where $\Sigma_y := \sigma^2 I_n + \Phi\Gamma\Phi^T$ and $\Gamma := \mathrm{diag}(\gamma)$. Then, we estimate γ by maximizing the marginalized probability distribution function. This is equivalent to minimizing $-\log p(y; \gamma)$, giving the cost function

$$\mathcal{L}_\gamma(\gamma) = \log|\Sigma_y| + y^T\Sigma_y^{-1}y, \tag{2.10}$$

that is,

$$\mathcal{L}_\gamma(\gamma) = \log\left|\sigma^2 I_n + \Phi\Gamma\Phi^T\right| + y^T\left(\sigma^2 I_n + \Phi\Gamma\Phi^T\right)^{-1}y. \tag{2.11}$$

On the other hand, for fixed hyperparameters, one can compute the posterior distribution of θ from the Bayes formula. Then, one has

$$p(\theta|y; \gamma) = \mathcal{N}(\mu_\theta, \Sigma_\theta), \tag{2.12}$$

with $\mu_\theta = \sigma^{-2}\Sigma_\theta\Phi^T y$ and $\Sigma_\theta = \left(\sigma^{-2}\Phi^T\Phi + \Gamma^{-1}\right)^{-1}$. After obtaining the estimate $\widehat{\gamma}$ of the hyperparameter vector, the optimal estimation of coefficient vector θ is

$$\widehat{\theta} = \widehat{\mu}_\theta = \left(\Phi^T\Phi + \sigma^2\widehat{\Gamma}^{-1}\right)^{-1}\Phi^T y = \widehat{\Gamma}\Phi^T\left(\sigma^2 I_n + \Phi\widehat{\Gamma}\Phi^T\right)^{-1}y, \tag{2.13}$$

where $\widehat{\Gamma} = \mathrm{diag}(\widehat{\gamma})$. Consider that

$$\begin{aligned} & y^T\left(\sigma^2 I_n + \Phi\Gamma\Phi^T\right)^{-1}y \\ & = \frac{1}{\sigma^2}y^T y - \frac{1}{\sigma^4}y^T\Phi\Sigma_\theta\Phi^T y = \frac{1}{\sigma^2}\|y - \Phi\mu_\theta\|^2 + \mu_\theta^T\Gamma^{-1}\mu_\theta \\ & = \min_{\theta}\left\{\frac{1}{\sigma^2}\|y - \Phi\theta\|^2 + \theta^T\Gamma^{-1}\theta\right\}. \end{aligned}$$

Thereby, we define the cost function with respect to θ and γ as follows:

$$\begin{aligned} \mathcal{L}(\theta, \gamma) &= \frac{1}{\sigma^2}\|y - \Phi\theta\|^2 + \theta^T\Gamma^{-1}\theta + \log\left|\sigma^2 I_n + \Phi\Gamma\Phi^T\right| \\ &\triangleq f(\theta, \gamma) + g(\theta, \gamma), \end{aligned} \tag{2.14}$$

where

$$\begin{aligned} f(\theta, \gamma) &= \frac{1}{\sigma^2}\|y - \Phi\theta\|^2 + \theta^T\Gamma^{-1}\theta, \\ g(\theta, \gamma) &= \log\left|\sigma^2 I_n + \Phi\Gamma\Phi^T\right|. \end{aligned}$$

It can be proven that $f(\theta, \gamma)$ is convex with respect to (θ, γ) and $g(\theta, \gamma)$ is concave with respect to (θ, γ).

The problem to determine the sparse coefficient θ is now equivalent to finding

$$(\widehat{\theta}, \widehat{\gamma}) \in \arg\min_{(\theta,\gamma)\in\Omega} \mathcal{L}(\theta, \gamma). \tag{2.15}$$

This is a DC (difference in convex functions) programming problem that can be generally solved by the CCP (concave–convex procedure) methods.

C. Convergence guarantee of the sparse Bayesian learning algorithm

The CCP-based SBL algorithm is first derived in Ref. 43, where an principled iterative reweighted ℓ_1 approach is designed to optimize the constructed auxiliary cost functions as shown in Eq. (2.14). Here, we take a slightly different formulation to present the used CCP-based SBL algorithm. Start with the CCP procedure, it is an iterative algorithm that solves the following sequence of convex programs,

$$\left(\theta^{(k+1)}, \gamma^{(k+1)}\right) \in \arg \min_{(\theta,\gamma)\in\Omega} \widehat{\mathcal{L}}\left(\theta, \gamma; \theta^{(k)}, \gamma^{(k)}\right), \tag{2.16}$$

or equivalently,

$$\left(\theta^{(k+1)}, \gamma^{(k+1)}\right) \in \arg \min_{(\theta,\gamma)\in\Omega} \left\{ \frac{1}{\sigma^2}\|y - \Phi\theta\|^2 + \sum_i \left(\frac{\theta_i^2}{\gamma_i} + c_i^{(k)}\gamma_i \right) \right\},$$

where

$$\begin{aligned} \widehat{\mathcal{L}}\left(\theta, \gamma; \theta^{(k)}, \gamma^{(k)}\right) &:= f(\theta, \gamma) + g\left(\theta^{(k)}, \gamma^{(k)}\right) \\ &\quad + \left\langle g_\gamma\left(\theta^{(k)}, \gamma^{(k)}\right), \gamma - \gamma^{(k)} \right\rangle, \\ c^{(k)} &:= \nabla_\gamma \log\left|\sigma^2 I_n + \Phi\Gamma^{(k)}\Phi^T\right|, \\ \Gamma^{(k)} &:= \operatorname{diag}\left(\gamma_1^{(k)}, \gamma_2^{(k)}, \dots, \gamma_m^{(k)}\right). \end{aligned}$$

Note that the supplementary definition for $\mathcal{L}$ can be extended to the function $\widehat{\mathcal{L}}$. It follows that

$$\theta^{(k+1)} \in \arg\min_\theta \left\{ \|y - \Phi\theta\|^2 + 2\sigma^2 \sum_i \sqrt{c_i^{(k)}}|\theta_i| \right\}, \tag{2.17a}$$

$$\gamma_i^{(k+1)} = \frac{|\theta_i^{(k+1)}|}{\sqrt{c_i^{(k)}}}, \; i = 1, 2, \dots, m. \tag{2.17b}$$

Given an initial point $\left(\theta^{(0)}, \gamma^{(0)}\right) \in \operatorname{int}(\Omega)$, we can obtain an iterative sequence $\left\{\left(\theta^{(k)}, \gamma^{(k)}\right)\right\}_{k=0}^{\infty}$. Moreover, $c^{(k)}$ can be analytically calculated by

$$c^{(k)} = \nabla_\gamma \log\left|\sigma^2 I_n + \Phi\Gamma^{(k)}\Phi^T\right| = \operatorname{diag}\left[\Phi^T\left(\sigma^2 I_n + \Phi\Gamma^{(k)}\Phi^T\right)^{-1}\Phi\right]. \tag{2.18}$$

By the property of concave function, one has

$$\mathcal{L}(\theta, \gamma) \le \widehat{\mathcal{L}}\left(\theta, \gamma; \theta^{(k)}, \gamma^{(k)}\right).$$

Combining with (2.16) yields

$$\begin{aligned} \mathcal{L}\left(\theta^{(k+1)}, \gamma^{(k+1)}\right) &\le \widehat{\mathcal{L}}\left(\theta^{(k+1)}, \gamma^{(k+1)}; \theta^{(k)}, \gamma^{(k)}\right) \\ &\le \widehat{\mathcal{L}}\left(\theta^{(k)}, \gamma^{(k)}; \theta^{(k)}, \gamma^{(k)}\right) = \mathcal{L}\left(\theta^{(k)}, \gamma^{(k)}\right). \end{aligned} \tag{2.19}$$

Then, some simple computations give

$$\mathcal{L}\left(\theta^{(k)}, \gamma^{(k)}\right) \ge \log|\sigma^2 I_n| = 2n\log\sigma > -\infty.$$

which shows that $\left\{\mathcal{L}\left(\theta^{(k)}, \gamma^{(k)}\right)\right\}_{k=0}^{\infty}$ is a bounded monotonic nonincreasing sequence. In fact, it is strictly monotonic before the iteration (2.17) reaches a local minimum. Therefore, the iterative process should be terminated when

$$\mathcal{L}\left(\theta^{(k)}, \gamma^{(k)}\right) - \mathcal{L}\left(\theta^{(k+1)}, \gamma^{(k+1)}\right) \le \tau, \tag{2.20}$$

where τ is a tolerance. The above iterative process is summarized in Algorithm 1.

ALGORITHM 1. SBL algorithm based on CCP.

```
Input: $\Phi \in \mathbb{R}^{n\times m}$: design matrix; $y \in \mathbb{R}^n$: observation vector; $\tau$: tolerance;
Output: $\theta^{(k+1)}$
initial$\Gamma^{(0)} = I_m$, $k = 0$;
repeat
    $c^{(k)} = \operatorname{diag}\left[\Phi^T\left(\sigma^2 I_n + \Phi\Gamma^{(k)}\Phi^T\right)^{-1}\Phi\right]$;
    $\theta^{(k+1)} \in \arg\min_\theta \left\{\|y - \Phi\theta\|^2 + 2\sigma^2\sum_i \sqrt{c_i^{(k)}}|\theta_i|\right\}$;
    $\gamma_j^{(k+1)} = \frac{|\theta_j^{(k+1)}|}{\sqrt{c_j^{(k)}}}$, $\Gamma^{(k+1)} = \operatorname{diag}\left(\gamma^{(k+1)}\right)$;
    $k = k + 1$
until $\left(\mathcal{L}\left(\theta^{(k)}, \gamma^{(k)}\right) - \mathcal{L}\left(\theta^{(k+1)}, \gamma^{(k+1)}\right) \le \tau\right)$
```

III. DISCOVERY OF COMPLEX GINZBURG–LANDAU EQUATIONS FROM BINARY-FLUID CONVECTION EXPERIMENT

In this section, we apply the proposed S^{3}d method to discover the complex Ginzburg–Landau equation (CGLE) from a binary-fluid convection experiment. Traveling-wave (TW) convection in binary fluids is a known, mainstream ansatz technique for studying the physical mechanism of non-equilibrium pattern-forming systems.[44] Substantial experiments on TW convection unfold many different dynamical states of TW. A major challenge in the scientific study of TW is quantitatively understanding and predicting the underlying dynamics. First-principle models based on the CGLE and its variants[45,46] have long been used to quantitatively explain experimental observations.

A. Experimental data description

The experimental data are collected from an experiment on convection in an ethanol/water mixture in a long, narrow, annular container which is heated from below.[19,34] The objective of the experiment is to probe into the counterpropagating wave packet formation regime. As reported in Refs. 34, 47, and 48, a regular, small-amplitude traveling-wave (TW) state is observed in this experiment, which consists of pairs of quasilinear wave packets which propagate around an annular cell in opposite directions, referred to as "left" and "right." In Fig. 2, we show the dynamics of the TW state for the bifurcation parameter ε (scaled by the characteristic time τ_0 defined below): $\varepsilon\tau_0 = 1.77 \times 10^{-3}$, where the left-TW rapidly moves to the left with small amplitudes [denoted by $A_L(x, t)$] and collides periodically with the right-TW [denoted by $A_R(x, t)$].

The data are the complex amplitudes, $A_L(x, t)$ and $A_R(x, t)$ of the TW state observed above onset ($\varepsilon = 0$), which are independently

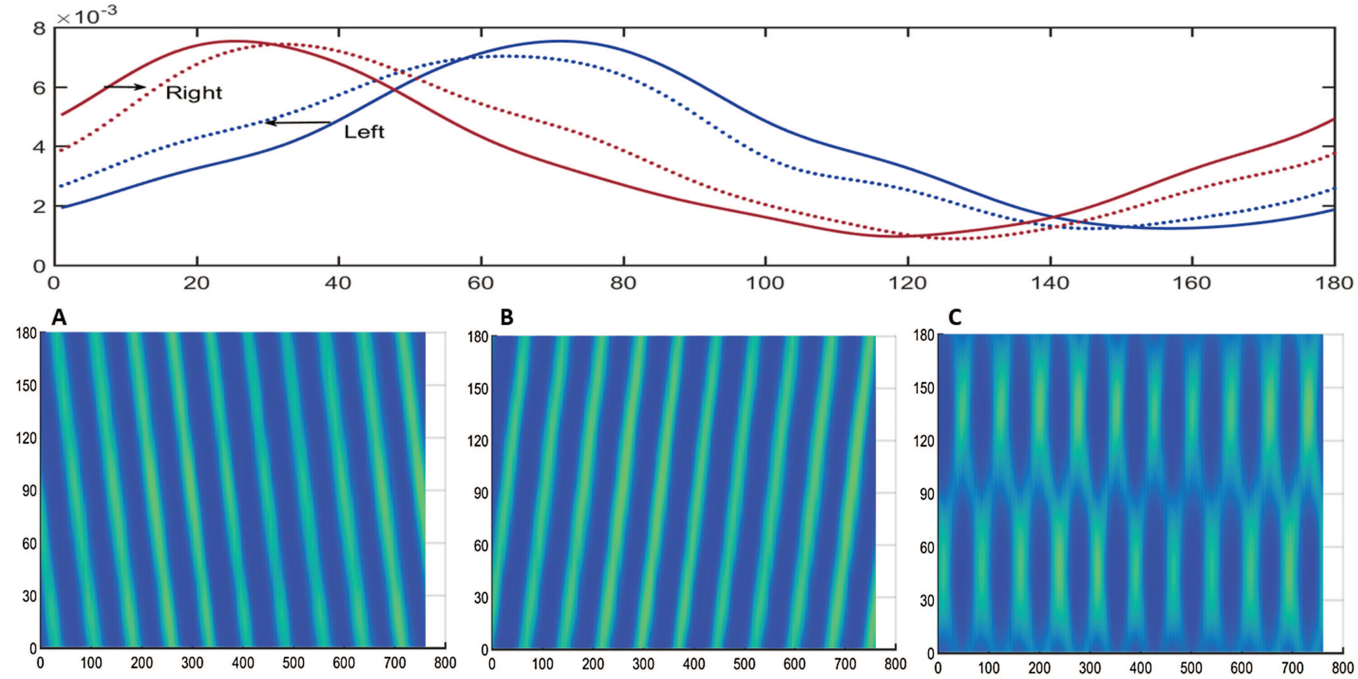


FIG. 2. The dynamics of the observed TW state for $\varepsilon\tau_0 = 1.77 \times 10^{-3}$. The top line shows the physical change of the amplitudes of the left-going (blue full line and dashed line) and right-going (red full curve and dashed line) wave components at a particular time and at the next time's data. (a) The behavior of left-going amplitudes; (b) the behavior of right-going amplitudes; (c) the behavior of the sum of left- and right-going amplitudes.

measured by applying full complex demodulation techniques[49] to the signals from the annular photodiode array, which records the flow-pattern image. The sampling period of these data is $\triangle t = 1.5625$, and the space sample interval is $\triangle x = 0.458\,167$. There are seven sets of data collected from the same experimental setup with seven different bifurcation parameters ε shown in Table I. For each dataset, a total of 180 × 760 data points is used for model discovery. Due to the use of the complex amplitudes, which is different from Ref. 19, where the real amplitudes and phases, $a_{L,R}(x,t)$ and $\phi_{L,R}(x,t)$ defined by $A_{L,R}(x,t) = a_{L,R}(x,t)e^{i\phi_{L,R}(x,t)}$, are used, Eq. (2.4) takes a complex form, that is, $\tilde{y} = \tilde{\Phi}\tilde{\theta}$, where $\tilde{y}, \tilde{\theta}$ are complex vectors and $\tilde{\Phi}$ is a complex matrix. For this, we consider the following one-to-one mapping from the complex vector $\tilde{y}$ to a real vector y and construct the following real matrix Φ:

$$y = \begin{bmatrix} \mathrm{Re}\,\tilde{y} \\ \mathrm{Im}\,\tilde{y} \end{bmatrix}, \quad \Phi = \begin{bmatrix} \mathrm{Re}(\tilde{\Phi}) & -\mathrm{Im}(\tilde{\Phi}) \\ \mathrm{Im}(\tilde{\Phi}) & \mathrm{Re}(\tilde{\Phi}) \end{bmatrix}.$$

TABLE I. Overview of the TW data in the designed experiment. The special letter "cal" and "car," respectively, represent the left-going waves and right-going ones. The files record the experimental data. Seven bifurcation parameters, ε, scaled by the characteristic time, τ_0 (i.e., $\gamma = \varepsilon\tau_0^{-1}$), represent the same experiments but conducted with different bifurcation parameters. All samples used for analysis are truncated from the 21st sampling point to the 780th sampling point[19] such that a total of 180 × 760 data points is used for the analysis. Then, we form the snapshot matrices, $U \in \mathbb{C}^{180\times760}$.

Experiment label	Bifurcation parameter	Sample number	Sampling interval	Space interval	Number of points after truncation
cal06172/car06172	$\gamma = 9.32 \times 10^{-3}$	988 × 180	1.5625	0.458 167	228
cal06182/car06182	$\gamma = 4.22 \times 10^{-3}$	800 × 180	1.5625	0.458 167	40
cal06192/car06192	$\gamma = 1.77 \times 10^{-3}$	1000 × 180	1.5625	0.458 167	240
cal06212/car06212	$\gamma = 6.38 \times 10^{-3}$	1000 × 180	1.5625	0.458 167	240
cal06222/car06222	$\gamma = 12.07 \times 10^{-3}$	1031 × 180	1.5625	0.458 167	271
cal06242/car06242	$\gamma = 14.03 \times 10^{-3}$	1000 × 180	1.5625	0.458 167	240
cal06252/car06252	$\gamma = 16.28 \times 10^{-3}$	1000 × 180	1.5625	0.458 167	240

With the above mapping and matrix construction, estimating the parameter $\tilde{\theta}$ can be equivalently converted to determine the θ via

$$\tilde{y} = \tilde{\Phi}\tilde{\theta} \Leftrightarrow y = \Phi\theta.$$

This one-shot optimization for complex variables can reduce the influence of the inherent noise on the discovery result. This is particularly important for model discovery due to the TW amplitudes are small (10^{-3} magnitude as shown in Fig. 2). This directly affects the computing error of the real amplitudes and phases by $a_{L,R}(x,t) = \sqrt{(\mathrm{Re}A_{L,R})^2 + (\mathrm{Im}A_{L,R})^2}$ as well as $\phi_{L,R} = \arctan \frac{\mathrm{Im}A_{L,R}}{\mathrm{Re}A_{L,R}}$ and aggravates the estimation error of their derivatives as demonstrated in Ref. 19.

B. Discovery of CGLE

We proceed the discovery process by two steps: first, the proposed S^{3}d method intensively determines the dominant physical terms from a large library of candidate terms. We construct the function dictionary using brief domain knowledge in fluid dynamics theory. Possible terms may include the terms appearing in the Navier–Stokes equations (such as $A \cdot \nabla A, \triangle A$) or the ones appearing in the first principles-based derived complex Ginzburg–Landau equations (such as the higher-order nonlinear terms $A^2\partial_x A, |A|^2 A$). In general, the amplitude itself and its derivative up to some custom order k should be added in order to properly model the physical system. Furthermore, to correctly model states of counterpropagating TW, the coupling terms like $|A_L|^2 A_R$ and $|A_R|^2 A_L$ must be added to account for the interactions between the left- and right-going TW. In this example, we take the third-order Volterra expansions of $A_{L,R}$ and $|A_{L,R}|$ and use the fourth-order compact Pade scheme[50] for the derivative estimation. It is shown that the Pade scheme is much more robust than the explicit fourth-order difference scheme for the noise. We investigate the performance of the S^{3}d method on the last three sets of data with a larger signal-to-noise ratio $\left(\text{that is, } \varepsilon\tau_0^{-1} \geq 12.07 \times 10^{-3}\right)$ in Table I. The second step is to identify coefficients for each identified dominant term, where all the data displayed in Table I are used for the coefficient estimations. The used candidate terms include A, $|A|$, A^2, $|A|^2$, A^3, $A^2|A|$, $A|A|^2$, $|A|^3$, A_{xx}, AA_x, AA_{xx}, A^2A_x, A^2A_{xx}, A^3A_x, A^3A_{xx}, and $|A_R|^2 A$ for the right-going TW.

The first finding is that the proposed S^{3}d method stably extracts five dominant feature terms to compose the following governing coupled equations:

$$\begin{aligned}
\partial_t A_R = {} & (0.0079 + 0.0640i)A_R - 0.5982\partial_x A_R \\
& + (0.9210 - 0.1243i)\partial_x^2 A_R \\
& + (17.7800 - 206.3442i)|A_R|^2 A_R \\
& + (-25.2965 - 150.6041i)|A_L|^2 A_R, \\
\partial_t A_L = {} & (0.0065 + 0.0443i)A_L + 0.5895\partial_x A_L \\
& + (0.3392 - 0.2185i)\partial_x^2 A_L \\
& + (-17.1600 - 89.0851i)|A_L|^2 A_L \\
& + (-6.4579 - 139.7492i)|A_R|^2 A_L,
\end{aligned} \tag{3.1}$$

where the estimated coefficients correspond to the experimental data labeled "cal/car06172." In Fig. 3, we show the relationship between the evolution of the fitting error and the change of the number of feature terms. It can be seen that for all used dataset, there exists a critical value before which the fitting-error slowly increases and after which the error begins to grow rapidly. The critical value to some extent illustrates that the retained five feature terms are best for describing the data.

In Fig. 4, we outline all the used candidate terms and enumerate the key feature terms for the bifurcation parameter, $\gamma = 14.03 \times 10^{-3}$, which include the amplitude itself A, $\partial_x A$, $\partial_{xx} A$, $|A|^2 A$ and the coupling term $|A_L|^2 A$ with the "R" subscript neglected. The same results also hold for two other datasets with bifurcation parameter $\gamma = 12.07 \times 10^{-3}$ and bifurcation parameter $\gamma = 16.28 \times 10^{-3}$. In fact, the discovered governing equations agree well with the simplest model of coupled complex Ginzburg–Landau equations (CGLEs) derived from first principles,

$$\begin{aligned}
\tau_0(\partial_t + s\partial_x)A = {} & \varepsilon(1 + ic_0)A + \omega_0^2(1 + ic_1)\partial_x^2 A \\
& + g(1 + ic_2)|A|^2 A + h(1 + ic_3)|A_L|^2 A,
\end{aligned}$$

where A is the complex amplitude of a right-going wave with group velocity s, parameter ω_0 is a characteristic length scale, and τ_0 is a characteristic time, which is determined experimentally by measuring the growth rate, $\gamma = \varepsilon\tau_0^{-1}$, at several values of ε and fitting the slope, $c_0 - c_3$ are dispersion coefficients, g is a nonlinear saturation parameter, and h is a nonlinear coupling coefficient, which reflects the stabilizing interaction between oppositely propagating traveling-wave components (see Ref. 19 and the reference therein). According to substantial theoretical[51–53] and experimental[47,48] works, the CGLE model has been proven to be potential to qualitatively explain weakly nonlinear dynamics near onset and closely parallel the experimental observations. Here, we apply the sparse Bayesian learning method to make a quantitive comparison of the experimental data with the simplest CGLE model.

Based on the analysis above, we further apply the proposed S^{3}d method to fine-tune the coefficients of CGLE. For TW convection, most parameters in CGLE (the coupling coefficient is an exception) have been measured in Ref. 35. Here, to show our identified results in an accessible form, we introduce the concept of the leave-one-out in machine learning to discuss the best method to choose the regularization parameters λ and obtain the best-fit coefficients. We accept the parameters that lead to the smallest fitting error on the test sets: $err = \frac{\|Y - \Phi\omega\|^2}{\|Y\|^2}$. In Table II, we summarized the identified coefficients, from which it can be seen that the estimated coefficients mostly resemble the theoretical values and experimental ones (Refs. 19, 34, and 35 and reference therein), except for some obscure parameters in theory and experiment. Particularly, we observe that for smaller bifurcation parameters, the estimated coupling coefficients possess the same characteristic, that is, $h\tau_0^{-1}$ for the right-going TW component A_R is smaller than ones for the A_L, and the value of $hc_3\tau_0^{-1}$ approximate -150. But for the three datasets with the highest signal-to-noise ratio, we clearly find that the estimated coefficients are unstable, which may arise from the bad estimates of the derivatives. Additionally, it is interesting to note that all the seven analyzed datasets have one thing in common, that

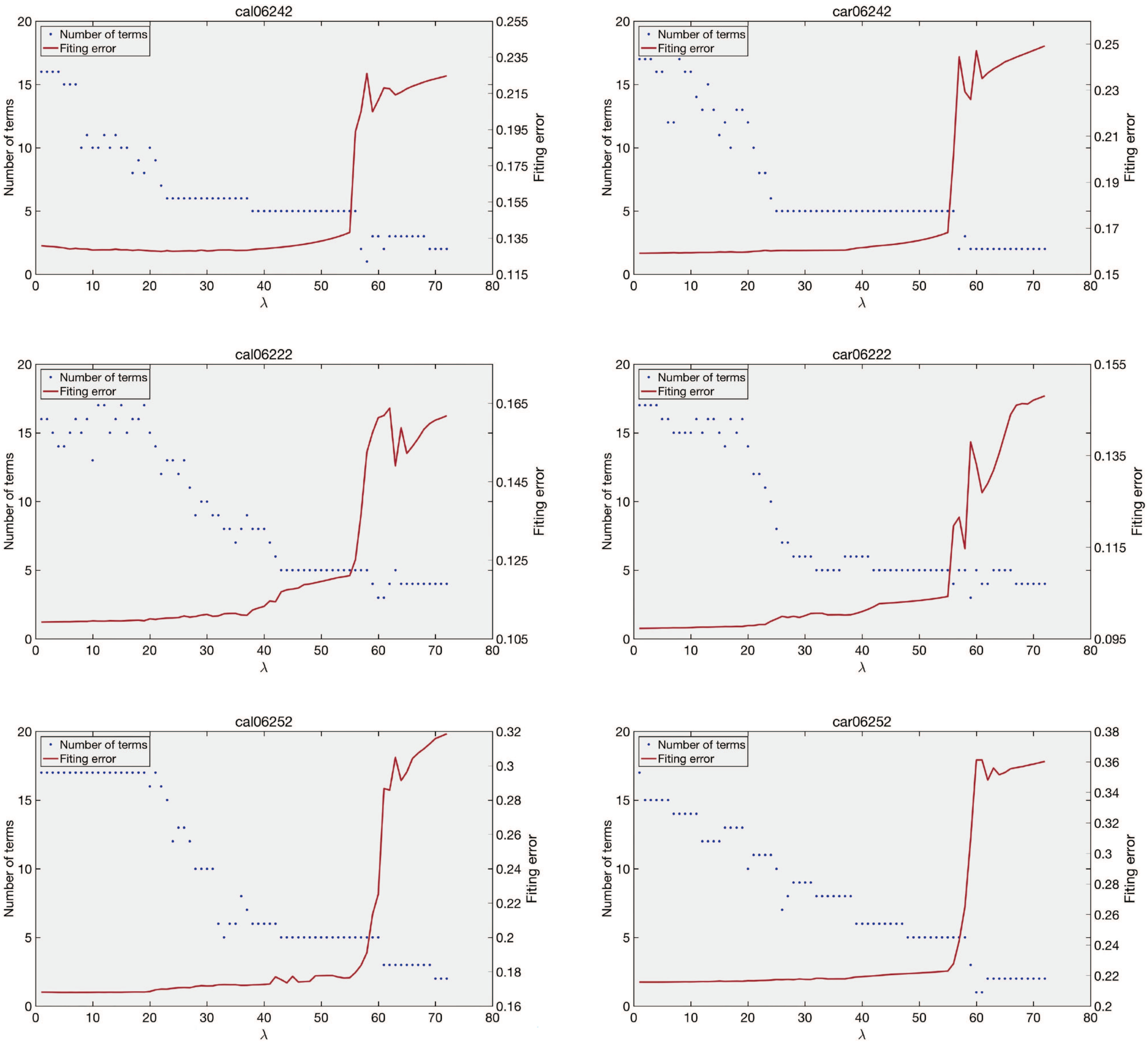


FIG. 3. The accelerations of fitting error due to a change in the number of feature terms. We empirically adjust the regulation parameter, λ, while observing the change in the fitting error. Three sets of data were chosen to test S^3d: car/cal06242, car/cal06222, and car/cal06252 from top to bottom. For each dataset, the fitting error slowly increased and tended to a relatively stable value [five features in Eq. (3.1) emerge], then that error began to rapidly grow.

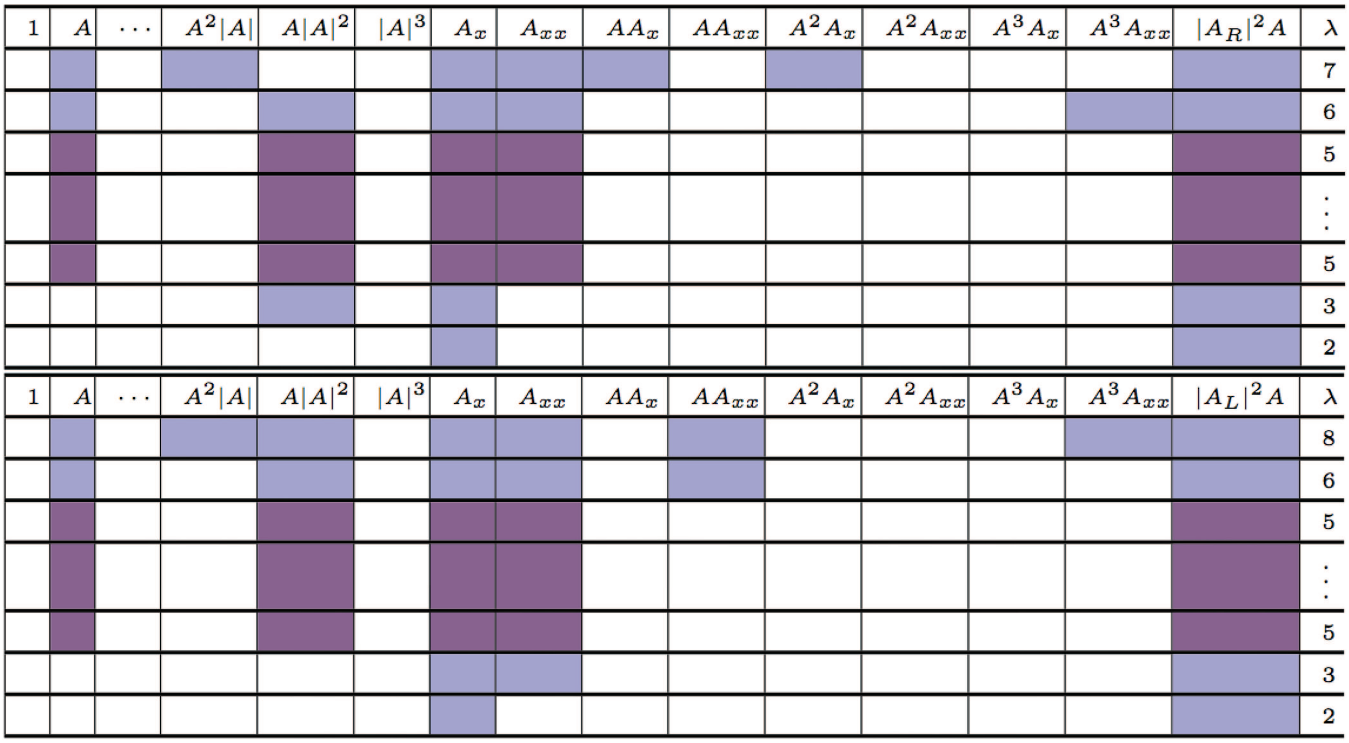
is, the negativity of the coupling coefficients, which indicates a stabilizing nonlinear competition between the two TW and a slowing down of the TW group velocity, which is also similar to Ref. 19 and agrees with the experimental observation.[34]

C. Model validation

One important benefit of data-driven PDE discovery is to use the discovered model for simulation, analysis, and prediction. Here, we make an attempt to solve the identified CGLE and reconstruct the TW convection behaviors with the appropriate initial boundary data. Note that the considered experiments are conducted in an annular container, which implies that the physical system comprises periodic boundary conditions,[19] that is,

$$A_L(x,t) = A_L(x+p,t), A_R(x,t) = A_R(x+p,t),$$

where p is one periodic. We employ the Fourier spectral method presented in Ref. 41 to simulate the identified CGLE model, for



1	A	$\cdots$	$A^2\lvert A\rvert$	$A\lvert A\rvert^2$	$\lvert A\rvert^3$	A_x	A_{xx}	AA_x	AA_{xx}	A^2A_x	A^2A_{xx}	A^3A_x	A^3A_{xx}	$\lvert A_R\rvert^2A$	λ
	░		░			░	░	░		░				░	7
	░			░		░	░						░	░	6
	▨			▨		▨	▨							▨	5
	▨			▨		▨	▨							▨	⋮
	▨			▨		▨	▨							▨	5
				░		░								░	3
						░								░	2

1	A	$\cdots$	$A^2\lvert A\rvert$	$A\lvert A\rvert^2$	$\lvert A\rvert^3$	A_x	A_{xx}	AA_x	AA_{xx}	A^2A_x	A^2A_{xx}	A^3A_x	A^3A_{xx}	$\lvert A_L\rvert^2A$	λ
	░		░	░		░	░		░				░	░	8
	░			░		░	░		░					░	6
	▨			▨		▨	▨							▨	5
	▨			▨		▨	▨							▨	⋮
	▨			▨		▨	▨							▨	5
						░	░							░	3
						░								░	2

FIG. 4. All the used candidate terms for the experimental data labeled car/cal06242, the proposed method identifies the feature terms with different λ.

TABLE II. Summary of the identified parameters of seven sets of real data: The data are labeled according to "right"(car) and "left"(cal). Each row represents the identified coefficients by S^{3}d.

	Coefficients in the discovered complex Ginzburg–Landau equation								
Experiment label	s	$\varepsilon\tau_0^{-1}$	$\xi_0^2\tau_0^{-1}$	$g\tau_0^{-1}$	$h\tau_0^{-1}$	$\varepsilon\tau_0^{-1}c_0$	$\xi_0^2c_1\tau_0^{-1}$	$gc_2\tau_0^{-1}$	$hc_3\tau_0^{-1}$
cal06172cb	0.5895	0.0065	0.3392	−17.1619	−6.4595	0.0443	−0.2185	−89.0851	−139.7500
car06172cb	0.5982	0.0079	0.9210	17.7800	−25.2965	0.0640	−0.1243	−206.3442	−150.6041
cal06182cb	0.6268	0.0057	0.0000	−78.0836	−0.8076	−0.0094	−0.3774	0.0000	−154.4844
car06182cb	0.6308	0.0081	0.0935	−90.2362	−35.0632	−0.0059	−0.4397	−55.8157	−157.3323
cal06192cb	0.5930	0.0054	0.0148	−139.4494	−2.5230	−0.0066	−0.6457	−37.6346	−152.7833
car06192cb	0.5916	0.0087	0.0158	−198.6799	−34.9115	−0.0066	−0.9189	−68.1182	−141.3105
cal06212cb	0.6063	0.0057	0.0639	−45.0095	−7.9602	0.0033	−0.2732	−35.3580	−154.5495
car06212cb	0.6131	0.0042	0.0202	−26.1553	−33.0555	0.0019	−0.1489	−17.7879	−168.7832
cal06222cb	0.5741	0.0173	1.7492	40.1657	−15.1882	0.0281	−0.0424	−379.5853	−136.5903
car06222cb	0.5771	0.0127	1.3000	29.9928	−16.6945	0.0114	−0.0894	−272.3181	−161.5083
cal06242cb	0.5509	0.0138	1.0701	16.2302	−17.3846	0.0555	−0.1505	−234.8376	−144.0718
car06242cb	0.5597	0.0150	1.1368	13.1359	−4.5687	0.0567	−0.2108	−237.9784	−149.1559
cal06252cb	0.5634	0.0199	1.1690	2.1061	−5.2084	0.0867	−0.2690	−267.3739	−169.0035
car06252cb	0.5481	0.0131	1.1075	22.3429	−3.9619	0.0794	−0.1245	−234.6182	−179.6777

instance, for the bifurcation parameter $\gamma = 6.38 \times 10^{-3}$,

$$\begin{aligned}\partial_t A_R &= -0.6131\partial_x A_R + (0.0042 + 0.0019i)A_R\\ &\quad + (0.0158 - 0.9189i)\partial_x^2 A_R\\ &\quad - (26.1553 + 17.7879i)|A_R|^2 A_R\\ &\quad - (7.9602 + 154.5495i)|A_L|^2 A_R,\\ \partial_t A_L &= 0.6063\partial_x A_L + (0.0057 + 0.0033i)A_L\\ &\quad + (0.0639 - i0.2723)\partial_x^2 A_L\\ &\quad - (45.0095 + 35.3580i)|A_L|^2 A_L\\ &\quad - (33.0555 + 168.7832i)|A_R|^2 A_L,\end{aligned}$$

where $0 < x < l, t > 0$, with the system length $l = 82.0119$ and the computing time $T = 625$ (i.e., $t_0 = 0, t_f = 625$). The simulation proceeds by discretizing the left-going complex wave amplitude, A_L, and the right-going complex wave amplitude, A_R, in space on a uniform 180 mesh with spectral approximations of the spatial derivatives, and integrating in time using an explicit Runge–Kutta formula with 401 time points. We use any column of the snapshot matrix, U, as an initial value. Figure 5 shows the simulation results for different initial values: the 622th column of U for data labeled "cal/car06172," the 600th column for data labeled "cal/car06192," the 600th column for data labeled "cal/car06212," and the 1st column for data labeled "cal/car06182." Therein, the first three rows have a resolution of 180×400 samples and the fourth row has a resolution of 180×200. Both resolutions correspond to reconstruction times $T = 625$ and $T = 312.5$, respectively. It can be seen that the TW behavior by simulation agrees well with the experimental observation. In simulation, we also find that the reconstruction results change with different initial values. One notable attempt is to take the 658th column for data labeled "cal/car06192" as the initial value. It can be observed that with the simulation running to $T = 1187$ both the left- and right-going TW possess regular bursts, and the amplitudes of wave approach the real values, as shown in Fig. 6. Thus, the reconstruction from our identified CGLE model bears a striking resemblance to the experimentally observed nonlinear states of TW convection on binary-fluid convection.

IV. DISCOVERY OF PROTOTYPICAL PDES

In this section, we validate the practicality of the proposed S^{3}d method by using various synthetic data from numerical solutions to canonical PDEs as well as the dataset originally collected in Ref. 24. Similar to Ref. 24, we add the Gaussian noise into the synthetic data for seeing the robustness of the SBL method. The error from the noisy data is quantified by means of the root mean square error (RMSE),

$$Err(y) = \sqrt{\frac{1}{n_x n_t}\sum_{j=1}^{n_x}\sum_{i=1}^{n_t}\left(u(x_j,t_i) - y(x_j,t_i)\right)^2}.$$

The mean-square error (MSE) and Standard Deviation (STD) are computed and used for assessing the discovery results. Additionally, we also use non-Gaussian noise to verify the robustness of the present method, which include the Gaussian mixture noise and the noise that stems from an Ornstein–Uhlenbeck process proposed in Refs. 54 and 55. Finally, we would like to mention that the assumption on the noise type for our theoretical analysis is unnecessary for the following experiments. The error analysis for data-driven PDE method is important, but which goes beyond our current research.

Table III demonstrates that S^{3}d method has successfully discovered the sine-Gordon equation, the Korteweg–de Vries equation, the FitzHugh–Nagumo equation, the Schrödinger's equation, and the Navier–Stokes equation. Due to page limit, we only demonstrate the discovery results for the Navier–Stokes equation.

A. Navier–Stokes equation

Consider the incompressible Navier–Stokes (NS) equations for the unsteady two-dimensional flows on torus with vorticity/stream function formulation,[36]

$$\begin{aligned}\omega_t + u\omega_x + v\omega_y &= (\omega_{xx} + \omega_{yy})/Re,\\ \psi_{xx} + \psi_{yy} &= -\omega,\end{aligned}\tag{4.1}$$

where $u = \psi_y$ represents the horizontal velocity component, $v = -\psi_x$ represents the vertical velocity component, $\omega = u_y - v_x$ represents the vorticity, and Re is the Reynolds number (based on the radius of the cylinder and the free-stream velocity, V_∞). The vorticity formulation is attractive for the accurate solution of high Reynolds number planar or axisymmetric NS equations.[56] In this example with the NS equations, the flow field is described by four field quantities, each of which needs to be measured for data.

We use a combination of the Fourier spectral method and the Crank–Nicolson method[56] to solve Eq. (4.1) with initial condition in $x \times y \in [0, 2\pi] \times [0, 2\pi]$,

$$\begin{aligned}\omega(x,y) &= \exp\left(-\frac{1}{5}\left[x^2 + \left(y + \frac{\pi}{4}\right)^2\right]\right)\\ &\quad + \exp\left(-\frac{1}{5}\left[x^2 + \left(y - \frac{\pi}{4}\right)^2\right]\right)\\ &\quad - \frac{1}{2}\exp\left(-\frac{2}{5}\left[\left(x - \frac{\pi}{4}\right)^2 + \left(y - \frac{\pi}{4}\right)^2\right]\right).\end{aligned}$$

We perform the spatial discretization on a uniform grid with $\triangle x = 0.0628$ and $\triangle y = 0.0628$. By substituting the Fourier approximation of the solution, u, v, ω, into Eq. (4.1), we further integrate the resultant ODEs using a time step of $\triangle t = 0.1$ using the Crank–Nicolson scheme. In Fig. 7, we plot the 2D NS pseudo-spectral solver on the torus with Reynolds number, $Re = 100$. The discrete version of the numerical solution forms the snapshot matrices, $U, V, W \in \mathbb{R}^{100\times100\times1001}$.

The dictionary matrix, Φ, consists of $m = 60$ basis functions. The underlying dynamics is described with the three state variables, u, v, w. Thus, our dictionary of basis functions include the linear combinations of the variables, $u, v, \omega, u_x, v_x, u_{xx}, v_{xx}$. With snapshots, U, V, W, we use the explicit difference scheme to approximate the first- or second-order derivatives in the noise-free case. We see from the discovery results that our method has strong performance with the noise-free dataset, a small MSE, and a small STD. We enumerate the re-sampling region in Table III.

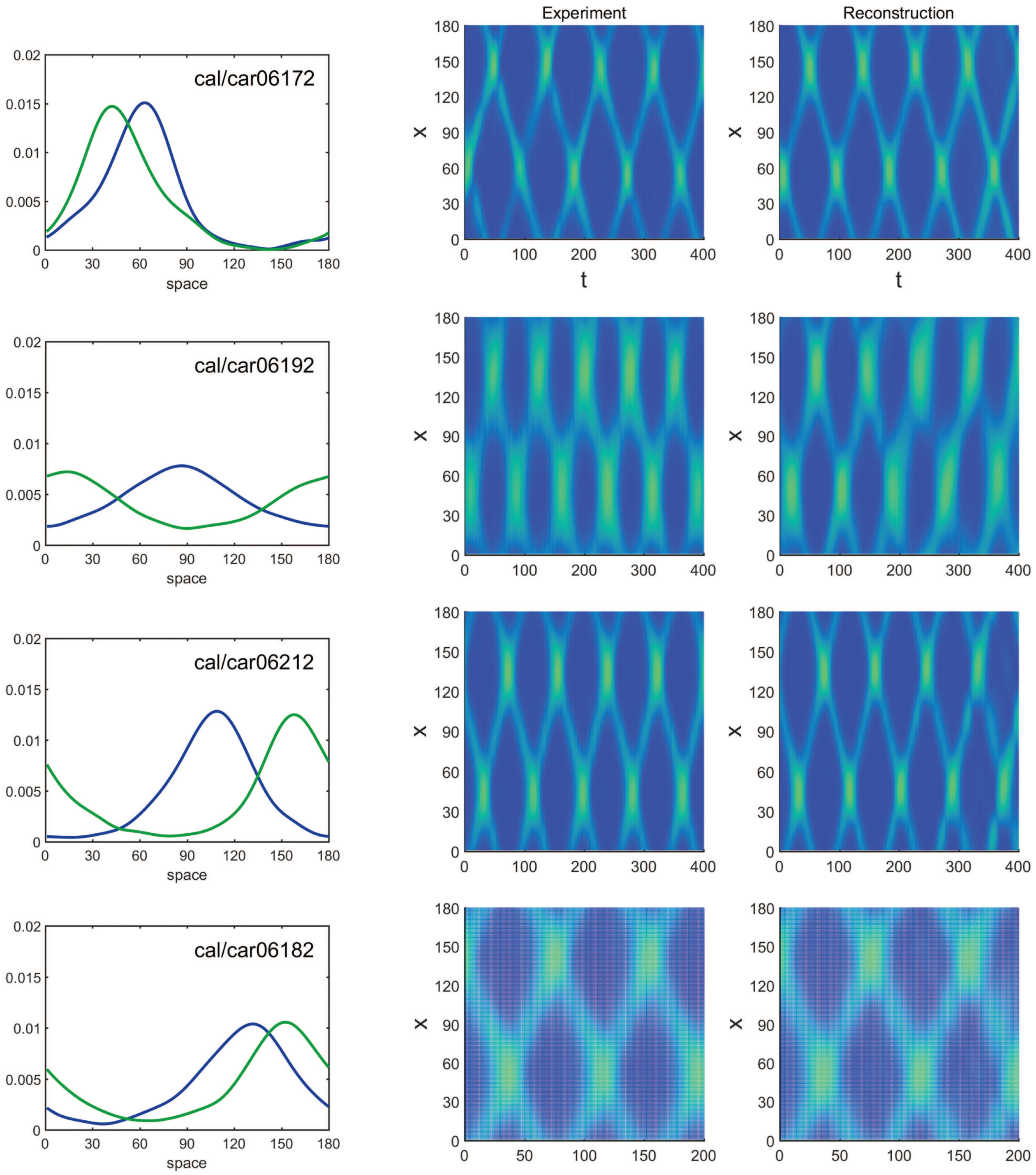


FIG. 5. Comparison of spatiotemporal evolution of the superposition of the left- and right-TW amplitude between the simulation and the experiment. The different initial values, the 622th column of the snapshot matrix U for data labeled "cal/car06172," the 600th column for data labeled "cal/car06192," the 600th column for data labeled "cal/car06212," and the 1st column for data labeled "cal/car06182," are selected for simulation (the first column). The first 400 (or 200) snapshots of the real data and the simulation data are shown with a resolution of 180 × 400 (column 2 and column 3).

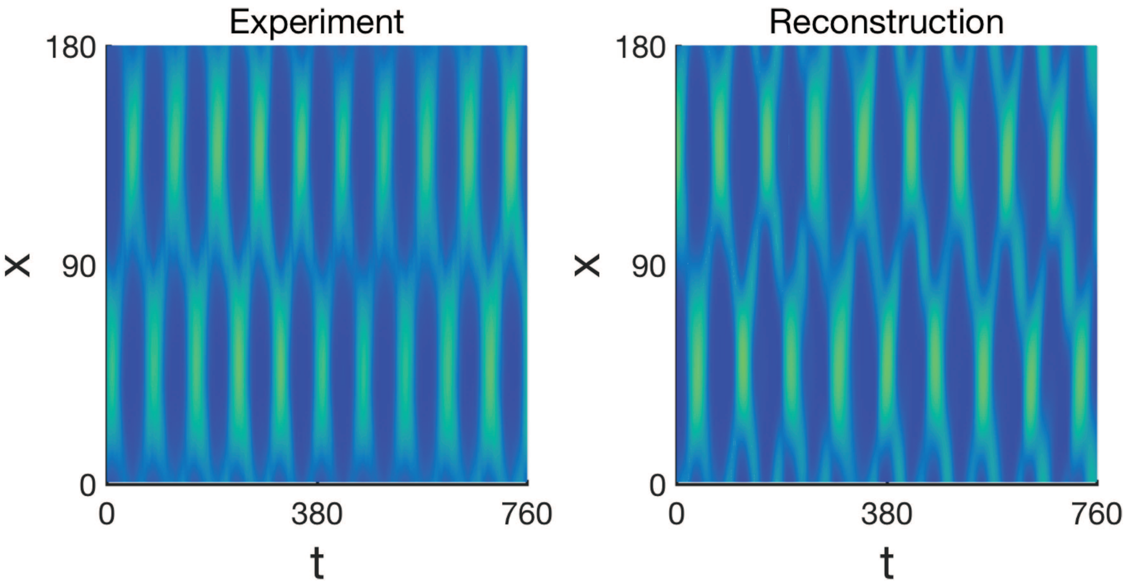


FIG. 6. Comparison of spatiotemporal evolution of the superposition of the left- and right- TW amplitude between the simulation and the experiment. Using the extended time $T = 1187$ and initial value (i.e., 658th column of snapshot matrix), a regular bust of TW is seen .

In the noisy case, we use the fourth-order compact Pade scheme with the S^{3}d method. We add 1% Gaussian noise to the original snapshot and the resulting RMSE is $Err = 0.4823$ for u, $Err = 0.4338$ for v, and $Err = 1.7111$ for w. We apply the POD method used in Refs. 57 and 58 to de-noise our data. We reshape the snapshot U into a $10\,000 \times 1001$ matrix, $\bar{U}$; we do the same shaping for W and V. By projecting the data onto 33 POD modes obtained using a threshold of 99.99 fixed, we generate a new snapshot, $\tilde{U} \in \mathbb{R}^{10000\times1001}$,

TABLE III. Summary of the discovery results for the sine-Gordon equation, the Korteweg–de Vries equation, the FitzHugh–Nagumo equation, the Schrödinger's equation, and the NS equation.

PDE	Noise case	Points	Identified PDE	Mean(err)±STD(err)
Sine-Gordon equation	No noise	10	$u_{tt} = 0.9999u_{xx} - 0.9986\sin(u)$	$0.0732\% \pm 0.0893\%$
	With noise	50	$u_{tt} = 0.9934u_{xx} - 0.9986\sin(u)$	$0.3999\% \pm 0.3738\%$
Korteweg–de Vries equation	No noise	10 000	$u_t = -0.000\,484u_{xxx} - 0.999\,246uu_x$	$0.0857\% \pm 0.0146\%$
	With noise	18 750	$u_t = -0.000\,474u_{xxx} - 0.974\,795uu_x$	$2.3186\% \pm 0.2856\%$
FitzHugh–Nagumo equation	No noise	10000 (u) 10 000 (w)	$u_t = -0.1989u + 1.1991u^2$ $-1.0010u^3 + 1.0023u_{xx} - 1.0005\omega$ $\omega_t = -0.000\,990\omega + 0.001\,998u$	$0.2952\% \pm 0.3502\%$
	With noise	18 000 (u) 27 000 (w)	$u_t = -0.1983u + 1.1812u^2$ $-0.9611u^3 + 1.0025u_{xx} - 0.9950\omega$ $\omega_t = -0.001\,000\omega + 0.001\,998u$	$1.0232\% \pm 1.3728\%$
Schrödinger's equation	No noise	10 000	$u_t = i0.3u_{xx} + i3.3333u\|u\|^2$ $-i3.3333u$	$0.0011\% \pm 0.0007\%$
	With noise	10 000	$u_t = i0.3013u_{xx} + i3.2946u\|u\|^2$ $-i3.3186u$	$0.6788\% \pm 0.4183\%$
Navier–Stokes equation	No noise	10 000	$\omega_t(x,t) = 0.0101\omega_{xx} + 0.0100\omega_{yy}$ $-1.0014u\omega_x - 0.9914v\omega_y$	$0.3830\% \pm 0.3781\%$
	With noise	20 000	$\omega_t(x,t) = 0.0101\omega_{xx} + 0.0097\omega_{yy}$ $-0.9988u\omega_x - 0.9996v\omega_y$	$1.0306\% \pm 1.5734\%$

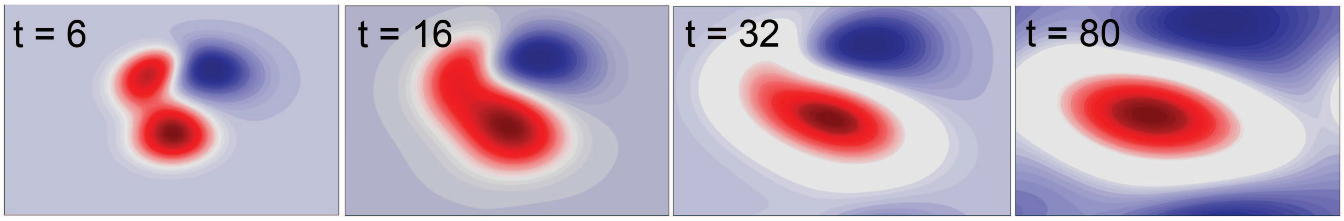


FIG. 7. Flow field for 2D NS equation (4.1) on the torus depicted in time $t = 6$, $t = 16$, $t = 32$, and $t = 80$.

from the matrix, $\bar{U}$: $\tilde{U} = \Psi A$ with Ψ consisting of 33 POD modes and the corresponding POD coefficients, A. Then, we reverse the process such that snapshots, $\bar{U}, \bar{V}, \bar{W} \in \mathbb{R}^{100\times 100\times 1001}$. The computed RMSE is 0.0056 for u, 0.0051 for v, and 0.0412 for w. We pay special attention to the Pade scheme. With the de-noised dataset, we approximate the first-order derivatives in each grid point and then we compute the second-order derivatives using the Pade scheme. After we compute the dictionary matrix, we correctly discover the NS equation as shown in Table III.

B. Comparison with SOTA methods on the prototypical examples from PDE-FIND[24]

We further verify the proposed S^{3}d method on the dataset originally collected in Ref. 24. Therein, these datasets are generated by simulating the KdV equation with single- and double-soliton solutions, the Burger's equation, the Quantum Harmonic Oscillator, and the nonlinear Schrödinger equation. Here, we neglect the simulation details and refer readers to Ref. 24. Table IV shows the identified results of applying the S^{3}d method to the dataset. It can be seen that the proposed S^{3}d method is predominant for small sample and obtains a better discovery of these PDEs in terms of a smaller parametric error. It is worthy noting that the dictionary matrices used for all PDEs in Table IV are different from Table III. For instance, the dictionary for the sine-Gordon equation is that

$$1, u, u^2, u^3, u^4, u_x, u_{xx}, u_{xxx}, uu_x, uu_{xx}, uu_{xxx},$$
$$u^2u_x, u^2u_{xx}, u^2u_{xxx}, u^3u_x, u^3u_{xx}, u^3u_{xxx}, u^4u_x, u^4u_{xx},$$
$$u^4u_{xxx}, \sin(u), \cos(u), \sin(u)\cos(u), \sin^2(u), \cos^2(u),$$

which are constructed by any combination of the state, $u(x,t)$, the corresponding derivatives up to the third-order and the periodic trigonometric functions. The dictionary constructed for the

TABLE IV. Comparison of identified results using PDE-FIND method and S^{3}d method with the original dataset in Ref. 24.

Example	S^{3}d method		PDE-FIND method	
Identified results	No. of samples	MSE(err)-STD(err)	No. of samples	MSE(err)-STD(err)
Korteweg–de Vries (single-soliton)				
$u_t = -4.9999u_x(c=5)$	750	0.0028% ± 0.0%	12 800	0.3745% ± 0.0%
$u_t = -0.9997u_x(c=1)$		0.0303% ± 0.0%		0.0820% ± 0.0%
$u_t = -6.0011uu_x - 1.0005u_{xxx}$	1500	0.0346% ± 0.0239%	25 600	2.5931% ± 1.3601%
Korteweg–de Vries (double-soliton)				
$u_t = -5.9909uu_x - 0.9995u_{xxx}$	9000	0.0995% ± 0.0742%	102 912	0.9572% ± 0.2322%
$u_t = -5.9250uu_x - 0.9867u_{xxx}$	16 600	1.2894% ± 0.0548%	85 432	7.4727% ± 4.9306%
Burgers' equation				
$u_t = -0.9999uu_x + 0.1000u_{xx}$	2000	0.0051% ± 0.0054%	25 856	0.1595% ± 0.0608%
$u_t = -1.0011uu_x + 0.1001u_{xx}$	5000	0.0878% ± 0.0323%	19 116	1.9655% ± 1.0000%
Quantum harmonic oscillator				
$u_t = 0.5000iu_{xx} - 1.0000i\dfrac{x^2}{2}u$	2000	0.0090% ± 0.0082%	205 312	0.2486% ± 0.0128%
$u_t = 0.4996iu_{xx} - 1.0005i\dfrac{x^2}{2}u$	2000	0.0693% ± 0.0269%	187 452	9.6889% ± 6.9705%
Nonlinear Schrödinger equation				
$u_t = 0.4998iu_{xx} + 1.0000i\|u\|^2u$	4500	0.0185% ± 0.0204%	256 512	0.0473% ± 0.0147%
$u_t = 0.5034iu_{xx} + 0.9987i\|u\|^2u$	7200	0.4096% ± 0.3934%	236 652	3.0546% ± 1.2193%

FitzHugh–Nagumo equation is that

$$\begin{gathered}1, u, u^2, u^3, u^3u, u_x, u_{xx}, u_{xxx}, uu_x, uu_{xx}, uu_{xxx},\\ u^2u_x, u^2u_{xx}, u^2u_{xxx}, u^3u_x, u^3u_{xx}, u^3u_{xxx},\\ u^4u_x, u^4u_{xx}, u^4u_{xxx}, w, w^2, w_x, ww_x, w^2w_x,\end{gathered}$$

which are constructed by any combination of the states, $u(x,t)$ and $w(x,t)$, and the corresponding derivatives up to the third order. The main difference between them is the number of states, the order of derivatives, and the degree of polynomials. In our proposed method, we always suggest that the constructed library is sufficiently large to have a sparse representation of the dynamics underlying the dataset, and the well-developed PDE theory can be integrated to add terms with physical meaning, such as convection terms and dissipative terms.

To understand the influence of the choice of dictionary matrix on discovery results, consider the one-dimensional Schrödinger equation with a cubic nonlinear term,

$$z_t = iz_{xx} + i\nu|z|^2 z, \tag{4.2}$$

where the Schrödinger operator iz_{xx} represents the advection (wave propagation) and the cubic nonlinear term $|z|^2z$ with the dissipation coefficient ν represents the diffusion. The actual data fields we analyze are the real representation and the imaginary one of z, denoted by $u(x,t)$ and $v(x,t)$, respectively. Inserting $z(x,t) = u(x,t) + iv(x,t)$ into (4.2), the dynamics of $u(x,t)$ and $v(x,t)$ are described by

$$\begin{aligned}u_t &= -v_{xx} - \nu v\left(u^2+v^2\right),\\ v_t &= u_{xx} + \nu u\left(u^2+v^2\right),\end{aligned}$$

which possess the traveling-wave solution with arbitrary constants c_1, c_2

$$\begin{aligned}u(x,t) &= c_1 \cos\left(x + \left(c_1^2\nu - 1\right)t + c_2\right),\\ v(x,t) &= c_1 \sin\left(x + \left(c_1^2\nu - 1\right)t + c_2\right).\end{aligned} \tag{4.3}$$

We take $c_1 = 5, \nu = 0.1, c_2 = 6$. In Fig. 8, we display the evolution of solution, where the uniform space grid possesses 512 elements and the time interval is $\triangle t = 0.003$. By computation, one can prove that the traveling-wave solution in (4.3) also satisfies

$$\begin{aligned}u_t &= -v_{xx} - 25\nu v,\\ v_t &= u_{xx} + 25\nu u.\end{aligned}$$

We construct a dictionary consisting of 25 basis functions. The basis functions include any combination of the state, $u(x,t)$, and the corresponding derivatives up to the third order. By applying the S^3d algorithm to the data, the identified results are as follows:

$$\begin{aligned}u_t &= 0.75v_{xx} - 0.75v,\\ v_t &= -0.75u_{xx} + 0.75u.\end{aligned}$$

It can be seen that our algorithm is prone to select the coefficients minimizing the ℓ_1 norm. Further, if the true term u_{xx} is not included in the candidate terms, the identified results are

$$\begin{aligned}u_t &= -1.5v,\\ v_t &= 1.5u,\end{aligned}$$

which demonstrates that our algorithm tends to select the parsimonious model.

We further verify the robustness of our S^3d algorithm by using the Gaussian mixture noise and the noise that stems from an Ornstein–Uhlenbeck process proposed in Refs. 54 and 55 to construct two new sets of noisy data. The Gaussian mixture noise obeys the

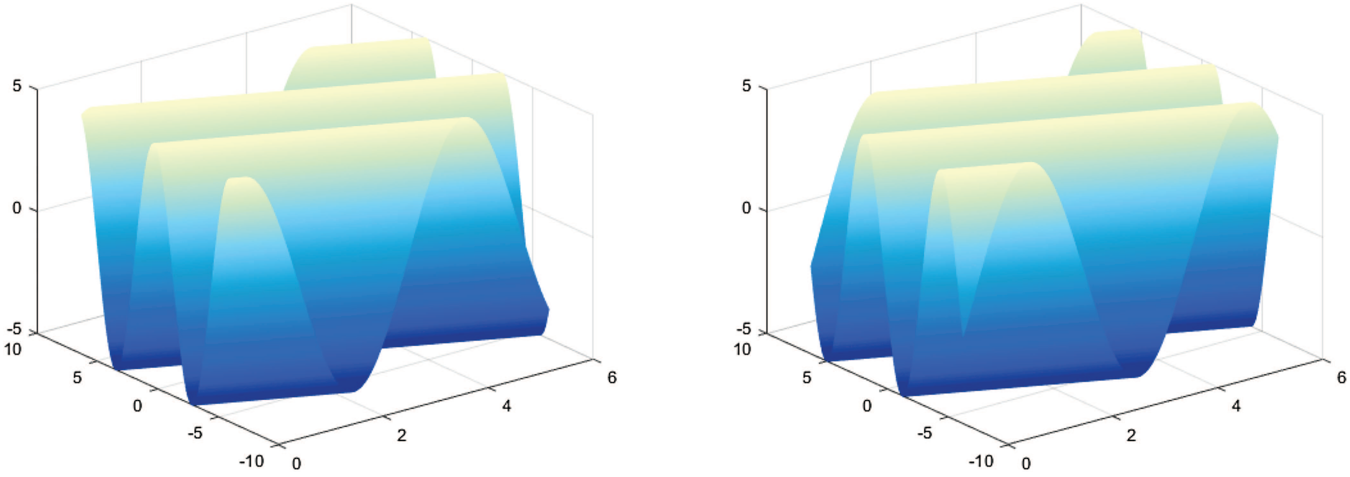


FIG. 8. Traveling solution of the Schrödinger equation (4.2).

following distribution:

$$(1-\xi)N\left(0,\,\sigma^2\right)+\xi N\left(0,\kappa\sigma^2\right),$$

where $\xi = 0.5$, $\sigma = 1$, and $\kappa = 2$. The magnitude of the noise is set to 1%. We randomly sub-sample 10 000 data points from the domain $[-0.9940, 0.2209] \times [2.56, 4.144]$. Our algorithm correctly discovers the nonlinear Schrodinger's equation with an accuracy of $MSE = 0.6907\%$, $STD = 0.4139\%$. The identified result is as follows:

$$u_t = i0.3014u_{xx} + i3.2944u|u|^2 - i3.3184u.$$

The noise stemming from an Ornstein–Uhlenbeck process violates the i.i.d. Gaussian noise. We write the measurements with added colored noise as

$$y\left(x_j, t_i\right) \;=\; u\left(x_j, t_i\right) \;+\; \eta_{j,i},$$

in which

$$\eta_{j,i} \;=\; \eta_{j,i-1} \;+\; \Delta t\theta\left(\mu - \eta_{j,i-1}\right) + \sqrt{\Delta t}\epsilon,$$

where Δt is the time stepsize, μ is the long-term expectation of the mean of noise, θ indicates the desire for the noise to migrate back toward the mean, and ϵ conforms to a normal distribution $\mathcal{N}\left(0,\,\sigma^2\right)$ with standard deviation σ. We set $\eta_{j,1} = 0$ at all spatial grid points at t_1, $\theta = 5$, $\mu = 0$, and $\sigma = 0.01$. We use the same snapshots and the basis functions. The experiment shows that our algorithm can correctly identify the nonlinear Schrödinger's equation from the data corrupted by the non-Gaussian noise. The experimental result is presented as follows:

$$u_t = i0.3000u_{xx} + i3.2809u\,|u|^2 - i3.3181u,$$

where mean = 0.6818%, STD = 0.8018%.

V. CONCLUSION AND DISCUSSION

This work proposes a framework to discover PDEs from experimental observations and synthetic spatiotemporal data. The proposed S^3d method discovers the dynamics underlying a state of traveling-wave convection from experimental data, i.e., the simplest complex Ginzburg–Landau equations. Additionally, many canonical PDEs are also re-discovered from synthetic spatiotemporal data. The merit of the proposed method is its ability to freely construct a model class with candidate functions and to automatically select the key ones that reproduce the observed spatiotemporal patterns. Benefiting from sparsity, the inferred PDEs are parsimonious and accurate, enabling interpretability. In the various examples, we observe that we are additionally able to robustly handle different types of noise and the measurement data with small sample size. We expect the S^3d method to be useful for the modeling of spatiotemporal dynamics from experimental data. This framework, as demonstrated through numerous examples, could potentially accelerate the discovery of new laws and stimulate physical explanations for the discovered equations, which lead to the discovery of the underlying mechanisms.

Although the dynamics we refer to in this work are Eulerian dynamics described by PDEs, S^3d, as a general method, is also able to infer ODEs and static functional relations using datasets in Refs. 24 and 59. S^3d unifies results for the discovery of natural laws.[5,59] Future work will focus on three important aspects of extension of our method for a wider range of practical applications. First, the currently proposed method relies on estimating temporal and spatial derivatives of the measured state variables from data, while there are cases that the system variables are not necessarily observable. Second, the current method can reconstruct equations that are linear with respect to parameters. We will extend this proposed method to the cases that are nonlinear in parameters. Finally, we assume that the constructed library is sufficiently large to have a sparse representation of the dynamics underlying the dataset. However, when there is no any prior knowledge on the to-be-identified governing equation, it is possible that the constructed library is insufficient. In these cases, genetic symbolic regression works much better. We may integrate genetic symbolic regression and the sparse regression into a unified framework, which could overcome this limitation.

ACKNOWLEDGMENTS

The authors acknowledge financial support from the National Natural Science Foundation of China under Grant Nos. 92167201 and 61903148.

We wish to thank Dr. Paul Kolodner for useful discussion and allowing us to use the experimental data.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Ye Yuan: Conceptualization (lead); Data curation (equal); Formal analysis (lead); Funding acquisition (lead); Investigation (lead); Methodology (lead); Project administration (lead); Resources (equal); Software (equal); Supervision (lead); Validation (equal); Visualization (equal); Writing – original draft (lead); Writing – review & editing (lead). **Xiuting Li:** Data curation (equal); Formal analysis (lead); Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Writing – review & editing (equal). **Liang Li:** Formal analysis (equal); Investigation (equal); Software (equal); Validation (equal). **Frank Jiang:** Formal analysis (equal); Investigation (equal); Software (equal); Validation (equal). **Xiuchuan Tang:** Formal analysis (equal); Investigation (equal); Validation (equal). **Fumin Zhang:** Formal analysis (equal); Project administration (equal); Supervision (equal). **Jorge Goncalves:** Formal analysis (equal); Project administration (equal); Writing – review & editing (equal). **Henning U. Voss:** Data curation (equal); Investigation (equal); Supervision (equal). **Han Ding:** Funding acquisition (equal); Project administration (equal); Supervision (equal). **Jürgen Kurths:** Data curation (equal); Methodology (equal); Project administration (equal); Resources (equal); Writing – review & editing (equal).

DATA AVAILABILITY

All synthetic data and codes used in this paper are publicly available in GitHub at https://github.com/HAIRLAB/S3d, Ref. 60.

APPENDIX: POLYNOMIAL APPROXIMATION

For noise-contaminated data, polynomial approximation is a better choice to alleviate effects due to noise. With sampled data, $u(x_j, t_i)$, at time, t_i, with $j = 1, \ldots, n_x$, we construct an approximation of the q-order derivative $\frac{\partial^q u(x,t)}{\partial x^q}$ by selecting the following sequence of polynomials of degree $p \in \mathbb{N}^+$ with $q < p$:

$$L_p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_p x^p,$$

subject to $L_p(x_j) = u(x_j, t_i)$. For example,

$$\begin{aligned} a_0 + a_1 x_1 + a_2 x_1^2 + \ldots + a_p x_1^p &= u(x_1, t_i), \\ a_0 + a_1 x_2 + a_2 x_2^2 + \ldots + a_p x_2^p &= u(x_2, t_i), \\ &\cdots\cdots \\ a_0 + a_1 x_{n_x} + a_2 x_{n_x}^2 + \ldots + a_p x_{n_x}^p &= u(x_{n_x}, t_i). \end{aligned} \tag{A1}$$

Further, we write Eq. (A1) into matrix form and solve for parameters using a QR factorization. Then, with $u(x, t_i) = L_p(x)$, we compute the qth-order derivative, $\frac{\partial^q u(x,t)}{\partial x^q}$. We demonstrate in our experiment that the error values caused by noise can be removed using polynomial approximation, leading a closer-to-real estimation of the derivative.

REFERENCES

[1] J. D. Farmer and J. J. Sidorowich, Phys. Rev. Lett. **59**, 845 (1987).

[2] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi, Phys. Rev. Lett. **106**, 154101 (2011).

[3] B. Dorneanu, S. Zhang, H. Ruan, M. Heshmat, R. Chen, V. S. Vassiliadis, and H. Arellano-Garcia, Front. Eng. Manag. **9**, 623 (2022).

[4] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidid, O. Runborg, and C. Theodoropoulos, Commun. Math. Sci. **1**, 715 (2003).

[5] J. Bongard and H. Lipson, Proc. Natl. Acad. Sci. **104**, 9943 (2007).

[6] Z. Shen, W.-X. Wang, Y. Fan, Z. Di, and Y.-C. Lai, Nat. Commun. **5**, 4323 (2014).

[7] I. Mezić, Annu. Rev. Fluid Mech. **45**, 357 (2013).

[8] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, J. Nonlinear Sci. **25**, 1307 (2015).

[9] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, J. Comput. Dyn. **1**, 391 (2014).

[10] D. Giannakis, Appl. Comput. Harmon. (2019).

[11] L. Ljung, *System Identification: Theory for the User* (Prentice-Hall, Inc., Upper Saddle River, NJ, 1986). ISBN 0-138-81640-9.

[12] A. Lindquist and G. Picci, *Linear Stochastic Systems: A Geometric Approach to Modeling, Estimation and Identification* (Springer, 2015), Vol. 1.

[13] D. Xu and O. Khanmohamadi, Chaos **18**, 043122 (2008).

[14] O. Khanmohamadi and D. Xu, Chaos **19**, 033117 (2009).

[15] M. Bär, R. Hegger, and H. Kantz, Phys. Rev. E **59**, 337 (1999).

[16] L. Guo and S. A. Billings, IEEE Trans. Circuits Syst. II: Express Briefs **53**, 657 (2006).

[17] T. Müller and J. Timmer, Int. J. Bifurcation Chaos **14**, 2053 (2004).

[18] L. Breiman and J. H. Friedman, J. Am. Stat. Assoc. **80**, 580 (1985).

[19] H. U. Voss, P. Kolodner, M. Abel, and J. Kurths, Phys. Rev. Lett. **83**, 3422 (1999).

[20] H. Voss, M. Bünner, and M. Abel, Phys. Rev. E **57**, 2820 (1998).

[21] R. Tibshirani, M. Wainwright, and T. Hastie, *Statistical Learning with Sparsity: The Lasso and Generalizations* (Chapman and Hall/CRC, 2015).

[22] W. Pan, Y. Yuan, J. Gonçalves, and G.-B. Stan, in *2012 IEEE 51st Annual Conference on Decision and Control (CDC)* (IEEE, 2012), pp. 2334–2339.

[23] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Proc. Natl. Acad. Sci. **113**, 312 (2016).

[24] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Sci. Adv. **3**, e1602614 (2017).

[25] H. Schaeffer, Proc. R. Soc. A **473**, 20160446 (2017).

[26] M. E. Tipping, J. Mach. Learn. Res. **1**, 211 (2001).

[27] A. C. Faul and M. E. Tipping, in *Advances in Neural Information Processing Systems* (The MIT Press, 2002), pp. 383–389.

[28] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado, IEEE Trans. Signal Process. **51**, 760 (2003).

[29] J. Palmer, K. Kreutz-Delgado, B. D. Rao, and D. P. Wipf, in *Advances in Neural Information Processing Systems* (The MIT Press, 2006), pp. 1059–1066.

[30] M. Figueiredo, in *Advances in Neural Information Processing Systems* (The MIT Press, 2002), pp. 697–704.

[31] D. P. Wipf and B. D. Rao, IEEE Trans. Signal Process. **52**, 2153 (2004).

[32] D. P. Wipf, B. D. Rao, and S. Nagarajan, IEEE Trans. Inf. Theory **57**, 6236 (2011).

[33] M. Raissi, P. Perdikaris, and G. E. Karniadakis, J. Comput. Phys. **378**, 686 (2019).

[34] P. Kolodner, Phys. Rev. Lett. **69**, 2519 (1992).

[35] P. Kolodner, S. Slimani, N. Aubry, and R. Lima, Physica D **85**, 165 (1995).

[36] C. Shu and B. E. Richards, Int. J. Numer. Methods Fluids **15**, 791 (1992).

[37] R. FitzHugh, Biophys. J. **1**, 445 (1961).

[38] E. Schrödinger, Phys. Rev. **28**, 1049 (1926).

[39] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems* (SIAM, 2007), Vol. 98.

[40] J. Li and Y.-T. Chen, *Computational Partial Differential Equations Using MATLAB* (Chapman and Hall/CRC, 2008).

[41] L. N. Trefethen, *Spectral Methods in MATLAB* (SIAM, 2000), Vol. 10.

[42] A. G. Baydin, B. A.Pearlmutter, A. A. Radul, and J. M. Siskind, J. Mach. Learn. Res. **18**, 5595 (2017).

[43] D. P. Wipf and S. S. Nagarajan, in *Advances in Neural Information Processing Systems* (The MIT Press, 2008), pp. 1625–1632.

[44] M. C. Cross and P. C. Hohenberg, Rev. Mod. Phys. **65**, 851 (1993).

[45] M. C. Cross, Phys. Rev. A **38**, 3593 (1988).

[46] A. C. Newell, in *Lectures in Applied Mathematics* (American Mathematical Society, Providence, RI, 1974), Vol. 15, p. 237.

[47] P. Kolodner, J. A. Glazier, and H. Williams, Phys. Rev. Lett. **65**, 1579 (1990).

[48] P. Kolodner, Phys. Rev. A **44**, 6466 (1991).

[49] P. Kolodner and H. Williams, in *Proceedings of the NATO Advanced Research Workshop on Nonlinear Evolution of Spatio-temporal Structures in Dissipative Continuous Systems* (Springer Science & Business Media, 1990).

[50] G. Strang, *Computational Science and Engineering* (Wellesley-Cambridge Press, Wellesley, 2007), Vol. 791.

[51] S. Linz and M. Lücke, Phys. Rev. A **35**, 3997 (1987).

[52] W. Schöpf and W. Zimmermann, EPL (Europhys. Lett.) **8**, 41 (1989).

[53] W. Schöpf and L. Kramer, Phys. Rev. Lett. **66**, 2316 (1991).

[54] G. E. Uhlenbeck and L. S. Ornstein, Phys. Rev. **36**, 823 (1930).

[55] D. J. Higham, SIAM Rev. **43**, 525 (2001).

[56] G. Strang, *Computational Science and Engineering* (Wellesley-Cambridge Press, Wellesley, 2007), Vol. 791.

[57] G. Berkooz, P. Holmes, and J. L. Lumley, Annu. Rev. Fluid Mech. **25**, 539 (1993).

[58] L. Sirovich, Q. Appl. Math. **45**, 561 (1987).

[59] M. Schmidt and H. Lipson, Science **324**, 81 (2009).

[60] Y. Yuan, *et al.* (2018) Matlab code for 'Machine discovery of partial differential equations from spatiotemporal data *Github.* https://github.com/HAIRLAB/S3d