J. Comput. Phys. 494 (2023) 112526

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Group Projected subspace pursuit for IDENTification of variable coefficient differential equations (GP-IDENT)

Yuchen He [a,*], Sung Ha Kang [b], Wenjing Liao [b], Hao Liu [c], Yingjie Liu [b]

[a] Department of Mathematics, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
[b] School of Mathematics, Georgia Institute of Technology, 686 Cherry St NW, GA 30332, Atlanta, GA, USA
[c] Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, 852, Kowloon, Hong Kong

ARTICLE INFO

Keywords:
PDE identification
Data-driven method
Model selection
Sparse regression

ABSTRACT

We propose an effective and robust algorithm for identifying partial differential equations (PDEs) with space-time varying coefficients from the noisy observation of a single solution trajectory. Identifying unknown differential equations from noisy data is a difficult task, and it is even more challenging with space and time varying coefficients in the PDE. The proposed algorithm, GP-IDENT, has three ingredients: (i) we use B-spline bases to express the unknown space and time varying coefficients, (ii) we propose Group Projected Subspace Pursuit (GPSP) to find a sequence of candidate PDEs with various levels of complexity, and (iii) we propose a new criterion for model selection using the Reduction in Residual (RR) to choose an optimal one among a pool of candidates. The new GPSP considers group projected subspaces which is more robust than existing methods in distinguishing correlated group features. We test GP-IDENT on a variety of PDEs and PDE systems, and compare it with the state-of-the-art parametric PDE identification algorithms under different settings to illustrate its outstanding performance. Our experiments show that GP-IDENT is effective in identifying the correct terms from a large dictionary, and our model selection scheme is robust to noise.

1. Introduction

Partial Differential Equations (PDEs) are indispensable and ubiquitous mathematical models articulating fundamental laws that govern various phenomena in physics, chemistry, biology, finance, and many other fields. Let the variable of interest be $u(x,t) : \Omega \times [0, T_{\max}] \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^d$ is a d-dimensional spatial domain, and $T_{\max} > 0$ is the final time of the observation. An important class of models that describe the dynamical features of u is the evolution PDE [8,4,38]

$$u_t = \mathcal{F}(u, \partial_x u, \partial_x^2 u, \cdots) \tag{1}$$

with a functional $\mathcal{F}$. In the multidimensional case with $d > 1$, the spatial location is given by $x = (x_1, \dots, x_d)$. We use the multi-indexing notation $\partial_x^m u = \{\partial_x^\alpha u := \partial_{x_1}^{\alpha_1} \partial_{x_2}^{\alpha_2} \cdots \partial_{x_d}^{\alpha_d} u \,, \alpha = (\alpha_1, \dots, \alpha_d) \,, \alpha_1 + \cdots + \alpha_d = m\}$ to denote the collection of m-order partial derivatives of u. The model (1) covers a wide range of important PDEs including the advection-diffusion equation for transferring physical quantities, the Kolmogorov-Petrovsky-Piskunov (KPP) equation [47] for population genetics, the incompressible Navier-Stokes equation [28], the

* Corresponding author. The author list is alphabetized.
E-mail addresses: royhe2@cityu.edu.hk (Y. He), kang@math.gatech.edu (S.H. Kang), wliao60@gatech.edu (W. Liao), haoliu@hkbu.edu.hk (H. Liu), yingjie@math.gatech.edu (Y. Liu).

https://doi.org/10.1016/j.jcp.2023.112526
Received 25 April 2023; Received in revised form 18 September 2023; Accepted 25 September 2023

Available online 29 September 2023
0021-9991/

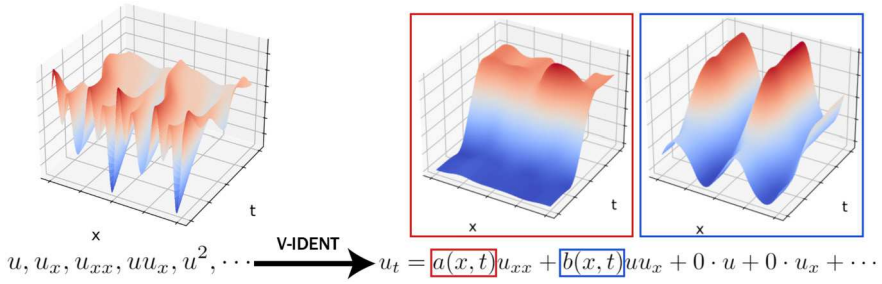


Fig. 1. The proposed algorithm, GP-IDENT, identifies the underlying PDE with space-time varying coefficients from a single trajectory of noisy observations.

Korteweg-de Vries (KdV) equation [33], and the Kuramoto-Sivashinsky (KS) equation [22] for fluid dynamics. For a vector-valued u, (1) also covers PDE systems such as the nonlinear Schrödinger equation [53] for light propagation. Model (1) can be regarded as an infinite dimensional dynamical system whose asymptotic properties such as attractors [4] and chaotic behaviors [38] have been extensively studied.

Classical approaches to derive PDEs for specific physical processes are based on physical laws and simplified assumptions. In modern science, **data-driven PDE identification** is explored to automatically identify such model (1) from the given observation. Such approaches allow scientists and engineers to discover non-linear and high-order complicated PDEs which are hard to model by empirical experience. In literature, various techniques have been developed to identify the active features, where sparse regression is one of the major frameworks for PDE identification [2,42,40,19,16,25,29,31,39,24,55,45]. Brunton et al. [2] studied the application of L_1-norm regularization in the context of PDE identification and proposed the sequential thresholded least-squares to find the active features. Kang et al. [19] proposed to obtain a series of candidate models using LASSO [46] and then select the optimal model with the minimal time evolution error (TEE). Rudy et al. [40] penalized the coefficients using the L_0-norm, and proposed sequential threshold ridge regression (STRidge) to solve the resulting problem. He et al. [9] proposed to use Subspace Pursuit (SP) [6] with a series of sparsity levels to generate candidate models. They also proposed Successively Denoised Differentiation (SDD) for denoising the input, and cross-validation error evaluation and multi-shooting TEE for selecting the optimal candidate. Other sparsity promoting penalties are studied in [20,5,3]. A theoretical analysis for PDE identification can be found in [12,54,11]. Methods such as [30,31,45] used sequential least squares [30,31] and subspace pursuit [45] on a weak form of PDE instead of differential form which are more robust to noise. Another line of works is based on neural networks [27,50,51,37], where sparse regression is embedded for feature selection, and a sufficient amount of trajectories of data are required for training. Different frameworks such as symbolic regression [1,43,48,26] are also available. See [34] for a recent review.

PDEs with space and time varying coefficients are often used in real applications, such as optimal control [49,14,21], trajectory planning [32], studies of piezoelectricity [17], and electromagnetic eddy current problems [23]. In such cases, certain coefficients may depend on both time and space, and in some equations, parts of the coefficients may vary with time while the others vary with space. An effective and robust PDE identification scheme with the flexibility of handling space and time varying coefficients is in need. The goal of PDE identification in this paper is to find an expression of (1) in a parametric form

$$u_t = \sum_{g=1}^{G} C_g(x,t) f_g(x,t) \tag{2}$$

based on single, possibly noisy, observations of a solution trajectory u in $\Omega \times [0, T_{\max}]$. The set of potential features $\mathcal{G} = \{f_g\}_{g=1}^{G}$ forms a dictionary, which can include linear terms such as partial derivatives of u in various orders, and products of multiple linear terms, e.g., uu_x and u^2. The size of the dictionary $G > 0$ is sufficiently large, and C_g, $g = 1, 2, \ldots, G$, represents a space-time dependent function. Fig. 1 provides an illustration: from the noisy observation of a single solution trajectory, the proposed method identifies the features u_{xx} and uu_x from a dictionary and reconstructs the respective space and time varying coefficients, i.e., $a(x,t)$ and $b(x,t)$.

There are few works dealing with space-time varying coefficients: [19,39] laid out a framework to identify varying coefficients and explored regularizers to encourage structural sparsity. Algorithms were numerically tested on PDEs with either space or time varying coefficients. In [19], the authors explored identification of spatially varying coefficients with Group-Lasso and proposed a Base Element Expansion (BEE) technique. In [10], the authors proposed a split Bregman method to identify interacting kernels in aggregation equations, where the kernel to be identified is space and time varying, yet the form of the equation is assumed to be given.

In this paper, we propose Group Projected subspace pursuit for IDENTification of variable coefficient PDEs (GP-IDENT) to identify parametric PDEs with space-time varying coefficients from a single trajectory of noisy data. After spanning the hypothesis space by B-spline bases [44], we generate a collection of candidate models by using different levels of group sparsity, then evaluate each candidate by considering the Reduction in Residual error (RR) to identify the optimal model. Since the candidate generation involves solving a non-convex, non-differentiable, NP-hard problem [6], we design a novel and effective Group Projected Subspace Pursuit (GPSP) greedy algorithm to produce candidate models with any specified level of group sparsity. We compare these methods on a variety of linear, non-linear PDEs and systems of differential equations with different levels of noise. Our experiments show that GP-IDENT outperforms other methods in terms of effectiveness, efficiency, and robustness. Contribution can be summarized as follows.

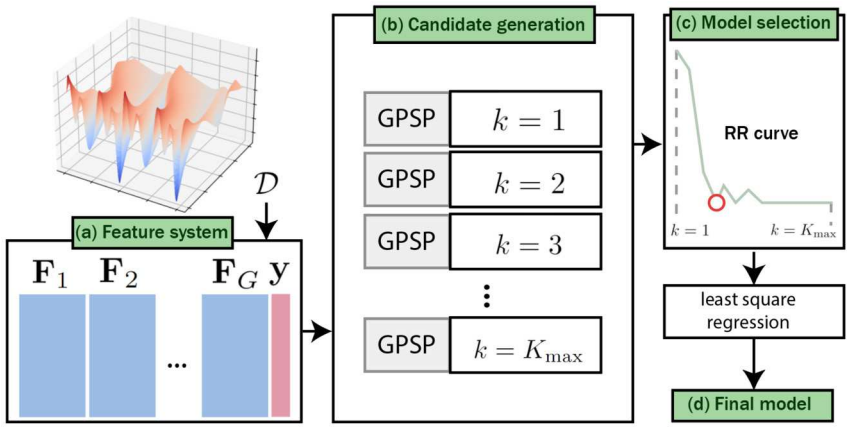


Fig. 2. Workflow of the proposed GP-IDENT for varying coefficient PDE identification from noisy observations. (a) Given the noisy observation of a single solution trajectory, we build the feature system (Section 2.1). (b) For $k = 1, \dots, K_{\max}$, we generate a candidate model by solving a group-ℓ_0 optimization problem, the proposed GPSP algorithm (Section 3). (c) We evaluate each candidate's Reduction in the Residual error (RR) to select the optimal model among candidate models. (Section 2.3). (d) Reconstruct the coefficients by least square regression.

1. We propose a novel method, GP-IDENT, to identify parametric PDEs with variable coefficients which vary in space and time. We assume that the given data are noisy observations of a single solution trajectory. The proposed procedure integrating SDD, GPSP, and RR shows robust performances compared to other state-of-the-art approaches.
2. We propose a new Group Projected Subspace Pursuit algorithm, GPSP, for structured sparse regression with group ℓ_0-norm constraint. GPSP is efficient in searching for the correct features in the underlying PDE, and outperforms block subspace pursuit [18] especially when different groups or columns within a group are highly correlated.
3. We propose the Reduction in the Residual error (RR) to identify the optimal model. This criterion gives more stable identification results compared to AIC-based approaches [39,24] when the data are noisy or the dictionary is large.

This paper is organized as follows. In Section 2, we present the detailed procedure of the proposed method, GP-IDENT. In Section 3, we describe the new Group Projected Subspace Pursuit algorithm, GPSP, and explain the details, including comparisons with block subspace pursuit [18]. Following numerical implementation details in Section 4, we present numerical experiments to validate the effectiveness of the proposed GP-IDENT and compare it with the state-of-the-art methods on various types of PDEs in Section 5. We conclude the paper with some discussions in Section 6.

2. Group Projected subspace pursuit for IDENTification (GP-IDENT) of variable coefficient differential equations

The proposed method has four steps as illustrated by the flowchat in Fig. 2. **[Step 1]** From the noisy observation of a single solution trajectory, to amend the instability caused by noise, we employ the Successively Denoised Differentiation (SDD) [9] to smooth the given data and generate the feature system as in [39,19]. Each variable coefficient is represented by B-spline bases [44] to account for the variation in space and time. The details are presented in subsection 2.1. **[Step 2]** To find candidate models of each sparsity level, we propose GPSP. We describe the procedure of GPSP in subsection 2.2 and the algorithmic details are presented in Section 3. **[Step 3]** Among the candidate models, we present the model selection criterion based on a Reduction in the Residual error (RR), detailed in subsection 2.3. **[Step 4]** Finally the coefficients are reconstructed. We summarize the proposed GP-IDENT algorithm in Algorithm 1.

Notation: In this paper, we use standard letters such as u, B for scalars. We use bold lowercase letters such as $\mathbf{c}$ for vectors, and bold uppercase letters such as $\mathbf{A}$ for matrices. For a matrix $\mathbf{A}$, $\mathbf{A}^\top$ denotes its transpose, and $\mathbf{A}^\dagger$ denotes its pseudo-inverse. A vector $\mathbf{c} \in \mathbb{R}^N$ is viewed as a column vector, and $\mathbf{c}^\top$ as its transpose is a row vector. $\|\mathbf{c}\|_1 = \sum_{n=1}^N |c_n|$ and $\|\mathbf{c}\|_2 = \sqrt{\sum_{n=1}^N c_n^2}$ are ℓ_1 and ℓ_2-norm of $\mathbf{c}$, respectively. We use $\mathrm{supp}(\mathbf{c}) := \{n = 1, 2, \dots, N \,|\, c_n \neq 0\}$ for the set of indices of the non-zero entries of $\mathbf{c}$, and its ℓ_0-norm $\|\mathbf{c}\|_0$ is the number of elements in $\mathrm{supp}(\mathbf{c})$.

2.1. Setup: construction of the feature system

In the first step, we set up a feature system for feature identification and coefficient reconstruction. To simplify the notations, we focus on one-dimensional spatial domain in the description.

Consider the evolution PDE in (1) on the spatial and temporal domain $\mathbb{S}^1 \times [0, T_{\max}]$ with a periodic boundary condition in space. Denote the noisy observations of its solution trajectory by

$$\mathcal{D} = \{U(x_i, t_n) = u(x_i, t_n) + \varepsilon_{i,n},\ i = 1, \dots, I,\ n = 1, \dots, N\}.$$

Algorithm 1 The proposed GP-IDENT algorithm.

Require: Sampled trajectory data $\mathcal{D}$, over-complete dictionary $\mathcal{G}$, smoothing window size $w \geq 0$, hypothesis space $\mathcal{H}_M$, maximal sparsity level $K_{\max}$, threshold ρ, and selection window L

1: Construct the feature system $(\mathbf{A}, \mathbf{y})$ based on $\mathcal{D}$, $\mathcal{H}_M$, and $\mathcal{G}$ using SDD with window size w.
2: **for** $k = 1, \dots, K_{\max}$ **do**
3: Obtain an approximate solution $\overline{\mathbf{c}}^*(k)$ with GPSP (Section 3) using $\overline{\mathbf{A}}$ and $\overline{\mathbf{y}}$, which are normalized $\mathbf{A}$ and $\mathbf{y}$, respectively.
4: **end for**
5: Compute s_k in (12) for $k = 1, \dots, K_{\max} - L$, and select the optimal candidate with sparsity k^* in (13).
6: Obtain $\mathbf{c}^*(k^*)$ by least square regression using partial columns of $\mathbf{A}$

$$\min_{\mathbf{c}} \|\mathbf{A}\mathbf{c} - \mathbf{y}\|_2^2 \text{ subject to } \operatorname{supp}(\mathbf{c}) = \operatorname{supp}(\overline{\mathbf{c}}^*(k^*)),$$

or simply rescale $\overline{\mathbf{c}}^*(k^*)$ according to the norms of columns of $\mathbf{A}$ and $\mathbf{y}$.
7: **return** A PDE model specified by $\mathbf{c}^*(k^*)$.

Here $\varepsilon_{i,n}$ is the noise. We assume that the underlying PDE is in the form of (2), i.e., it is a linear combination of features, e.g., u_x and uu_x, contained in an over-complete dictionary $\mathcal{G} = \{f_g : \mathbb{S}^1 \times [0, T_{\max}] \mapsto \mathbb{R}\}_{g=1}^G$ with coefficients that may depend on space and time. Note that (2) can represent nonlinear PDEs if $\mathcal{G}$ includes nonlinear features.

Let $\{B_m(x,t)\}_{m=1}^M$ be a set of bases, and denote $\mathcal{H}_M = \operatorname{Span}(\{B_m(x,t)\}_{m=1}^M)$ as a hypothesis space. We first approximate each variable coefficient $C_g(x_i, t_n)$ by an expansion of the basis elements such that

$$C_g(x_i, t_n) \approx \sum_{m=1}^{M} c_{g,m} B_m(x_i, t_n) \in \mathcal{H}_M$$

with constant coefficients $c_{g,m} \in \mathbb{R}$ for $g = 1, 2, \dots, G$. Then each term in (2) is represented as

$$C_g(x_i, t_n) f_g(x_i, t_n) \approx \sum_{m=1}^{M} c_{g,m} B_m(x_i, t_n) f_g(x_i, t_n), i = 1, \dots, I, \ n = 1, \dots, N. \tag{3}$$

Since the exact value of $f_g(x_i, t_n)$ is unknown, we approximate it by the empirical counterpart $\widehat{f}_g(x_i, t_n)$ estimated from the given data $\mathcal{D}$, which is detailed in Section 4. We express (3) in the matrix form:

$$C_g(x_i, t_n) f_g(x_i, t_n) \approx \mathbf{f}_g^\top(i, n) \mathbf{c}_g, \tag{4}$$

where

$$\mathbf{f}_g^\top(i, n) = \begin{bmatrix} \widehat{f}_g(x_i, t_n) B_1(x_i, t_n) & \cdots & \widehat{f}_g(x_i, t_n) B_M(x_i, t_n) \end{bmatrix} \in \mathbb{R}^M \tag{5}$$

and $\mathbf{c}_g = \begin{bmatrix} c_{g,1} & c_{g,2} & \dots & c_{g,M} \end{bmatrix}^\top \in \mathbb{R}^M$. Define the g-th group feature as

$$\mathbf{F}_g = \begin{bmatrix} \mathbf{f}_g(1,1) & \mathbf{f}_g(2,1) & \cdots & \mathbf{f}_g(I,N) \end{bmatrix}^\top \in \mathbb{R}^{IN \times M}.$$

We concatenate $\{\mathbf{F}_g\}_{g=1}^G$ to construct the **feature matrix**:

$$\mathbf{A} = \begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 & \cdots & \mathbf{F}_G \end{bmatrix} \in \mathbb{R}^{IN \times GM}, \tag{6}$$

as illustrated in Fig. 2 (a). Similarly, we construct $\mathbf{c}$ from $\mathbf{c}_g$ via

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_1^\top & \mathbf{c}_2^\top & \cdots & \mathbf{c}_G^\top \end{bmatrix}^\top \in \mathbb{R}^{GM}. \tag{7}$$

We approximate $u_t(i, n)$ by its empirical counterpart $\widehat{u}_t(i, n)$ based on the given data $\mathcal{D}$. We define the **feature response** as

$$\mathbf{y} = \begin{bmatrix} \widehat{u}_t(x_1, t_1) & \widehat{u}_t(x_2, t_1) & \cdots & \widehat{u}_t(x_I, t_N) \end{bmatrix}^\top \in \mathbb{R}^{IN}, \tag{8}$$

and refer to the pair $(\mathbf{A}, \mathbf{y})$ as a **feature system** derived from the given data $\mathcal{D}$ using the dictionary $\mathcal{G}$ and the hypothesis space $\mathcal{H}_M$.

2.2. Candidate generation using GPSP

In the second step, we generate a sequence of candidate models with distinct levels of sparsity. Let $K_{\max}$ be a fixed integer such that $1 \leq K_{\max} \leq G$. For $k = 1, 2, \dots, K_{\max}$, we consider

$$\min_{\overline{\mathbf{c}} \in \mathbb{R}^{GM}} \|\overline{\mathbf{A}}\overline{\mathbf{c}} - \overline{\mathbf{y}}\|_2^2 \text{ subject to } \|\overline{\mathbf{c}}\|_{\ell_{0,1}} = k, \tag{9}$$

where $\overline{\mathbf{A}}$ and $\overline{\mathbf{y}}$ are obtained from $\mathbf{A}$ and $\mathbf{y}$ by normalizing each column, i.e., the column norms are 1, and the $\ell_{0,1}$-norm of a vector $\mathbf{c} \in \mathbb{R}^{GM}$

$$\|\mathbf{c}\|_{\ell_{0,1}} := \left\| \begin{bmatrix} \|\mathbf{c}_1\|_1 & \dots & \|\mathbf{c}_G\|_1 \end{bmatrix} \right\|_0 \tag{10}$$

represents the number of groups with non-zero coefficients. The constraint enforces group sparsity by explicitly specifying that only k groups of features have nonzero coefficients. The solution of (9) corresponds to a PDE model with exactly k features that best fits the given data. However, due to the ℓ_0-norm constraint, exactly solving the non-convex and non-differentiable problem (9) is NP-hard [6].

We propose Group Projected Subspace Pursuit (GPSP) to find a group k-sparse vector $\overline{\mathbf{c}}^*(k)$ for $k = 1, 2, \dots, K_{\max}$. Given a fixed k, the proposed GPSP iteratively searches for k groups highly correlated to the residuals in a greedy manner (See Section 3). For each sparsity level k, we denote the index set corresponding to the active group features by $T(k) \subseteq \{1, \dots, G\}$. We obtain $K_{\max}$ candidate PDEs whose active features are indexed by $T(k)$ for each sparsity level $k = 1, \dots, K_{\max}$ respectively.

2.3. Model selection by Reduction in Residual (RR)

The third step is to select the optimal model from candidates specified by each sparsity level k. We design a new score using the sum of squared residuals (SSR)

$$R_k = \|\overline{\mathbf{A}}\overline{\mathbf{c}}^*(k) - \overline{\mathbf{y}}\|_2^2, \tag{11}$$

and compare the reduction of this residual for each k sparsity level. Let $L \geq 1$ be a fixed integer. For $k = 1, \dots, K_{\max} - L$, we compute the Reduction in Residual (RR) as

$$s_k = \frac{R_k - R_{k+L}}{L R_1}, \quad k = 1, \dots, K_{\max} - L. \tag{12}$$

This measures the average reduction of residual error as the sparsity level k increases. A small value in s_k means there is a marginal gain in accuracy as sparsity level gets bigger than k. Here, using $L = 1$ is not reliable: using GPSP, for each sparsity level k, the computation of (9) is totally independent. The index set $T(k)$ of the active features for the k-th candidate may not be a subset of $T(k+1)$, i.e., $R_k - R_{k+1}$ may be negative. By using the average of L in (12), we suppress the impact of fluctuation and improve the stability of model selection.

When the value s_k is already small, we choose the smallest sparsity k, rather than choosing k with the smallest s_k. We introduce a threshold $\rho > 0$, and pick the optimal sparsity as follows:

$$k^* = \min\{k : 1 \leq k \leq K_{\max} - L, s_k < \rho\}. \tag{13}$$

This is the smallest sparsity index k where s_k is below ρ. The motivation of this criterion is to find the simplest model, where RR does not reduce further by considering more complex models.

We find that GP-IDENT is not sensitive to the choice of $L > 1$ and ρ, and we fix $L = 5$ and $\rho = 0.015$ in this paper. We illustrate the effect of RR with an example in Appendix B.

2.4. Reconstruction of the coefficients

In the fourth step, we reconstruct the coefficients of the identified PDE. After obtaining the optimal level of sparsity k^* in Step 3, we reconstruct the coefficients $\mathbf{c}^*(k^*)$ by solving

$$\min_{\mathbf{c} \in \mathbb{R}^{GM}} \|\mathbf{A}\mathbf{c} - \mathbf{y}\|_2^2 \ \text{ subject to } \operatorname{supp} \mathbf{c} = \operatorname{supp} \overline{\mathbf{c}}^*(k^*), \tag{14}$$

where we recall that $\overline{\mathbf{c}}^*(k^*)$ is the approximate solution of (9) given by GPSP with the optimal group sparsity k^* selected in subsection 2.3. (14) is equivalent to a least square regression using the group features indexed by $T(k^*)$. Alternatively, we can reconstruct $\mathbf{c}^*(k^*)$ by properly rescaling $\overline{\mathbf{c}}^*(k^*)$ by the norms of columns of $\mathbf{A}$ and $\mathbf{y}$. In particular, the m-th entry $\mathbf{c}^*(k^*)$ is equal to the m-th entry of $\overline{\mathbf{c}}^*(k^*)$ divided by the norm of the m-th column of $\mathbf{A}$ then multiplied by the norm of $\mathbf{y}$.

3. Group Projected Subspace Pursuit (GPSP)

We propose the Group Projected Subspace Pursuit (GPSP) to generate candidates with k features. For Group-LASSO (GLASSO) [52] and the grouped version of STRidge, Sequential Grouped Threshold Ridge Regression (SGTR) [39], the sparsity level is implicitly controlled by a regularization parameter. GPSP allows one to explicitly specify the sparsity level, which makes the generation of the candidate models more efficient. Compared to Block Subspace Pursuit (BSP) [18], GPSP is numerically more stable when co-linearity occurs, and we show this in numerical experiments.

For the simplicity of notation, only in this section, we use $\mathbf{A}$ and $\mathbf{y}$ instead of $\overline{\mathbf{A}}$ and $\overline{\mathbf{y}}$, as the proposed GPSP is applicable in both cases and the normalization is used to make the algorithm numerically robust.

3.1. GPSP algorithm

For a fixed level of group sparsity $k \geq 1$, suppose the set of group indices selected by the $(l-1)$-th iteration is T^{l-1}, and denote as

$$\mathbf{y}_r^{l-1} = \operatorname{resid}(\mathbf{y}, \mathbf{A}_{T^{l-1}}) = \mathbf{y} - \operatorname{proj}(\mathbf{y}, \mathbf{A}_{T^{l-1}}) = \mathbf{y} - \mathbf{A}_{T^{l-1}} \mathbf{A}_{T^{l-1}}^{\dagger} \mathbf{y} \tag{15}$$

Algorithm 2 Group Projected Subspace Pursuit (GPSP) for (9).

Require: Feature system $(\mathbf{A}, \mathbf{y})$, specified level of group sparsity $k \geq 1$, maximal number of iterations $\text{Iter}_{\max} \geq 1$.

Set $l = 0$.
Set $T^l = \{k \text{ largest indices of } P(\mathbf{y}, \mathbf{F}_g), g = 1, 2, \ldots, G\}$ in (16).
Set $\mathbf{y}_r^l = \text{resid}(\mathbf{y}, \mathbf{A}_{T^l})$ in (15), $\mathbf{A}_{T^l}$ concatenates $\{\mathbf{F}_g\}_{g \in T^l}$ vertically.
for $l = 1, \ldots, \text{Iter}_{\max}$ **do**
  $\widetilde{T}^l = T^{l-1} \cup \{k \text{ largest indices of } P(\mathbf{y}_r^{l-1}, \mathbf{F}_g), g = 1, 2, \ldots, G\}$.
  Compute $\mathbf{x}_p^l = \mathbf{A}_{\widetilde{T}^l}^{\dagger} \mathbf{y}$.
  Set $T^l = \{k \text{ largest indices of } \|\mathbf{F}_g \mathbf{x}_p^l[g]\|_2,\ g \in \widetilde{T}^l\}$, where $\mathbf{x}_p^l[g]$ is the subvector of $\mathbf{x}_p^l$ corresponding to the g-th group.
  Compute $\mathbf{y}_r^l = \text{resid}(\mathbf{y}, \mathbf{A}_{T^l})$.
  if $\|\mathbf{y}_r^l\|_2 > \|\mathbf{y}_r^{l-1}\|_2$ **then**
    Set $T^l = T^{l-1}$ and terminate.
  end if
end for
return The optimal group indices T^l and the estimated coefficient $\mathbf{A}_{T^l}^{\dagger} \mathbf{y}$

the residual of fitting the data using groups specified by indices in T^{l-1}. Here $\mathbf{A}_{T^{l-1}}$ is obtained by concatenating the group features $\{\mathbf{F}_g\}_{g \in T^{l-1}}$ horizontally. The proposed scheme consists of two stages in each iteration: expanding and shrinking.

[Stage 1] Expand T^{l-1} to $\widetilde{T}^l$. For the l-th iteration, we first compute

$$P(\mathbf{y}_r^{l-1}, \mathbf{F}_g) = \frac{\left|\text{proj}(\mathbf{y}_r^{l-1}, \mathbf{F}_g)^{\top} \mathbf{y}_r^{l-1}\right|}{\|\text{proj}(\mathbf{y}_r^{l-1}, \mathbf{F}_g)\|_2 \|\mathbf{y}_r^{l-1}\|_2} \tag{16}$$

for $g = 1, 2, \ldots, G$. Note that $P(\mathbf{y}_r^{l-1}, \mathbf{F}_g)$ measures the correlation between $\mathbf{y}_r^{l-1}$ and its projection to the column space of $\mathbf{F}_g$. We take the union of T^{l-1} with the set of k groups with the highest k values in (16), and denote the union set as $\widetilde{T}^l$.

[Stage 2] Shrink $\widetilde{T}^l$ to T^l. Let $\mathbf{x}_p^l = \mathbf{A}_{\widetilde{T}^l}^{\dagger} \mathbf{y}$. We project $\mathbf{y}$ to the column space of $\mathbf{A}_{\widetilde{T}^l}$ with decomposition

$$\mathbf{y}_p = \text{proj}(\mathbf{y}, \mathbf{A}_{\widetilde{T}^l}) = \sum_{g \in \widetilde{T}^l} \mathbf{F}_g \mathbf{x}_p^l[g],$$

where $\mathbf{x}_p^l[g]$ is the subvector of $\mathbf{x}_p^l$ corresponding to the g-th group. For $g \in \widetilde{T}^l$, its norm $\|\mathbf{F}_g \mathbf{x}_p^l[g]\|_2$ provides a measure of the importance of the g-th group. Hence, from $\widetilde{T}^l$, we keep indices of k most important groups and remove the others. The refined set of indices is T^l.

After the l-th iteration, we compute $\mathbf{y}_r^l = \text{resid}(\mathbf{y}, \mathbf{A}_{T^l})$. If $\|\mathbf{y}_r^l\|_2 > \|\mathbf{y}_r^{l-1}\|_2$, we set $T^l = T^{l-1}$ and take the groups indexed by T^l as our final selection; otherwise, we repeat the procedure described above. We summarize the GPSP scheme in Algorithm 2.

3.2. Related algorithms

GPSP is closely related to Subspace Pursuit (SP) [6] and Block Subspace Pursuit (BSP) [18]. SP is a greedy algorithm for sparse regression. It iteratively expands the pool of k candidate covariates by considering potential features highly correlated to the residual, then refines the choices by reducing the extended pool back to k covariates by eliminating those with less importance. At each iteration, SP expands k nonzero entries to $2k$ nonzero entries by adding the k indices whose columns are highly correlated with the residual, and then refines the choice by eliminating the k indices with smaller coefficient values. From this perspective, both BSP and GPSP can be regarded as generalizations of SP where the covariates, i.e., individual columns of the system matrix, are replaced by groups of features. However, BSP and GPSP have different interpretations about the correlation between the residual and a feature group.

When expanding the pool of candidates from k to $2k$, BSP measures the correlation between the residual $\mathbf{y}_r$ and the g-th feature group $\mathbf{F}_g$ by the L_2-norm of the inner product between $\mathbf{y}_r$ and the columns of $\mathbf{F}_g$,

$$\|\mathbf{F}_g^T \mathbf{y}_r\|_2 = \sqrt{\sum_{m=1}^{M} (\mathbf{F}_g^{\top}[m] \mathbf{y}_r)^2}, \tag{17}$$

where $\mathbf{F}_g[m]$ denotes the m-th column of the g-th feature group. In GPSP, we use the inner product between $\mathbf{y}_r$ and its projection to the column space of $\mathbf{F}_g$ to quantify the correlation

$$P(\mathbf{y}_r, \mathbf{F}_g) = \frac{\left|(\mathbf{F}_g \mathbf{F}_g^{\dagger} \mathbf{y}_r)^T \mathbf{y}_r\right|}{\|\mathbf{F}_g \mathbf{F}_g^{\dagger} \mathbf{y}_r\|_2 \|\mathbf{y}_r\|_2}. \tag{18}$$

Comparing (17) with (18), we note that GPSP is less sensitive to co-linearity than BSP. If some columns of $\mathbf{F}_g$ are co-linear, BSP (17) considers that they all contribute to the correlation between $\mathbf{y}_r$ and $\mathbf{F}_g$, whereas GPSP (18) ignores the co-linear columns as they are redundant when representing the information contained in the group. See Fig. 3 for an illustration. Notice that if $\mathbf{F}_b$ only has one column, both (17) and (18) are identical to SP.

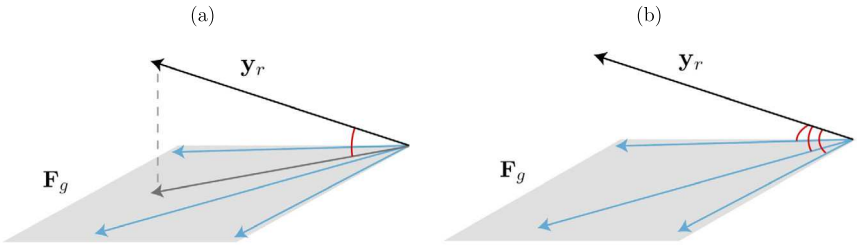


Fig. 3. An illustrative comparison between GPSP and BSP [18]. (a) In GPSP, the group's importance is evaluated by the correlation between $\mathbf{y}_r$ and its projection to the column space of $\mathbf{F}_g$. (b) In BSP, the group's importance is evaluated by the correlation between the residual $\mathbf{y}_r$ and the columns (blue arrows) in $\mathbf{F}_g$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 1
Comparison of two stages in BSP and GPSP (Algorithm 2). In [Stage 1] Expand (the first row), BSP chooses the groups $\mathbf{F}_g$ whose columns are highly correlated with the residual $\mathbf{y}_r^{l-1}$, whereas GPSP chooses the groups whose column spaces are close to the residual. In [Stage 2] Shrink (the second row), BSP selects the groups with large coefficients, while GPSP selects the groups whose projected residual is significant.

Criterion	BSP [18]	GPSP (Proposed)
Expand	$\|\mathbf{F}_g^\top \mathbf{y}_r^{l-1}\|_2$	$\left\|(\mathbf{F}_g\mathbf{F}_g^\dagger \mathbf{y}_r^{l-1})^\top \mathbf{y}_r^{l-1}\right\| / (\|\mathbf{F}_g\mathbf{F}_g^\dagger \mathbf{y}_r^{l-1}\|_2 \|\mathbf{y}_r^{l-1}\|_2)$
Shrink	$\|\mathbf{x}_p^l[g]\|_2$ for $g \in \widetilde{T}^l$	$\|\mathbf{F}_g \mathbf{x}_p^l[g]\|_2$ for $g \in \widetilde{T}^l$

When reducing the expanded pool of candidates of size $2k$ to k, BSP keeps the k groups whose reconstructed coefficients have the largest magnitudes, whereas GPSP uses each group's contribution measured by the norm of the response vector. Table 1 summarizes the differences between BSP and GPSP.

In general, GPSP is better suited for identifying PDEs with varying coefficients which are approximated by a basis expansion. As we allow the coefficients to vary both in space and time, some columns in the feature matrix can be highly correlated. We observe that GPSP is more effective than BSP when some columns within the same group are highly correlated. We illustrate this using the transport equation with constant speed $a \neq 0$, $u_t(x,t) = a u_x(x,t)$ in Appendix C. We numerically justify the advantages of GPSP over BSP with more examples in Section 5, and discuss computational efficiency of two methods in Appendix D.

4. Numerical implementation details

In this section, we present computational details for B-spine set-up and details of SDD used in this paper.

4.1. Approximation of varying coefficients by B-splines

For some fixed integer $M \geq 1$, we define $\mathcal{H}_M = \{\sum_{m=1}^M c_m B_m : c_m \in \mathbb{R}\}$ as our hypothesis space, where the basis function $B_m \in \mathbb{S}^1 \times [0, T_{\max}] \mapsto \mathbb{R}$ is compactly supported and $\sum_{m=1}^M B_m(x,t) = 1$ for all $(x,t) \in \mathbb{S}^1 \times [0, T_{\max}]$. The function space $\mathcal{H}_M$ is used to approximate the varying coefficients in the PDE, and we use the basis functions B_m's given by B-splines [36].

Without loss of generality, we consider $[0,1]$ as the spatial domain of interest. For a fixed integer $p \geq 1$, we consider a uniform knot sequence $0 = z_0 < z_1 < \cdots < z_l = 1$ for some $l \geq p$. Denote the knot spacing by Δz. The n-th B-spline basis function b_n^p of order p is constructed according to the Cox-de Boor recursion formula [7]

$$b_n^0(z) = \begin{cases} 1 & \text{if } z_n \leq z < z_{n+1}, \\ 0 & \text{otherwise}, \end{cases} \tag{19}$$

$$b_n^p(z) = \left(\frac{z - z_n}{z_{n+p} - z_n}\right) b_n^{p-1}(z) + \left(\frac{z_{n+p+1} - z}{z_{n+p+1} - z_{n+1}}\right) b_{n+1}^{p-1}(z), \tag{20}$$

for $0 \leq n \leq l - p - 1$. We note that b_n^p is non-zero on $[z_n, z_{n+p+1})$, and there are at most $p+1$ non-zero basis functions over any interval $[z_n, z_{n+1})$. Suppose the knot spacing is Δz, depending on different boundary conditions for the functions to be approximated, we supplement $\{b_n^p\}_{n=0}^{l-p-1}$ with more basis functions. For this purpose, it is convenient to uniformly extend the knot sequence to infinity $\cdots < z_{-2} < z_{-1} < z_0 < \cdots < z_l < z_{l+1} < \cdots$ where b_n^p is defined for $n \in \mathbb{Z}$.

- **Periodic boundary condition.** Add p functions $\widetilde{b}_n^p$ for $n = -p, -p+1, \ldots, -1$ defined as

$$\widetilde{b}_n^p(z) = \begin{cases} b_n^p(z) & \text{if } 0 \leq z < (n+p+1)\Delta z, \\ b_n^p(z-1) & \text{if } 1 + n\Delta z \leq z \leq 1, \\ 0 & \text{otherwise}. \end{cases} \tag{21}$$

- **Neumann boundary condition.** Add two functions

$$b_L^p(z) = \begin{cases} \sum_{n=-p}^{-1} b_n^p(z) & \text{if } 0 \le z < p\Delta z, \\ 0 & \text{otherwise,} \end{cases} \tag{22}$$

$$b_R^p(z) = \begin{cases} \sum_{n=l-p}^{l-1} b_n^p(z) & \text{if } 1 - p\Delta z \le z \le 1, \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

It is easy to check that $\sum_{n=0}^{l-p-1} b_n^p(z) + \sum_{n=-p}^{-1} \widetilde{b}_n^p(z) = 1$ and $\sum_{n=0}^{m-p-1} b_n^{p-1}(z) + b_L^p(z) + b_R^p(z) = 1$ when $z \in [0,1]$, and $\{b_n^p\}_{n=0}^{l-p-1} \cup \{\widetilde{b}_n^p\}_{n=-p}^{-1}$ serve as a set of B-spline basis functions of order p for $\mathbb{S}^1$.

In this paper, we assume periodic boundary condition in space. In the time direction, we assume the Neumann boundary condition so that the underlying coefficients do not have significant changes at the first nor the last moment of the observation. Suppose $\{b_{m_1}(x)\}_{m_1=1}^{M_1}$ is a set of B-spline bases constructed for $\mathbb{S}^1$, and another set $\{b_{m_2}(t)\}_{m_2=1}^{M_2}$ is constructed for $[0, T_{\max}]$ with supplementary elements for the Neumann boundary condition. We obtain a set of B-spline bases on the spatio-temporal domain $\mathbb{S}^1 \times [0, T_{\max}]$ by taking tensor products, that is,

$$B_m(x,t) \in \{b_{m_1}(x) b_{m_2}(t) : m_1 = 1, \dots, M_1,\ m_2 = 1, \dots, M_2\}$$

for $m = 1, 2, \dots, M$, where $M = M_1 M_2$.

Remark 1. Our algorithm can be adapted to handle different boundary conditions, provided that the feature terms in the dictionary can be computed accurately. One may consider computing partial derivatives with high accuracy on a truncated domain then interpolate to the entire domain, or use additional information of the coefficients to compute high order partial derivatives. The accuracy of the computed partial derivatives is important for the success of GP-IDENT and the accuracy of the recovered coefficients.

4.2. SDD for robust feature approximation

To robustly approximate f_g (respectively u_t) with $\widehat{f_g}$ (respectively $\widehat{u}_t$) using noisy observations of u (4), we suppress the noise amplification during the process of numerical differentiation. We apply the Successively Denoised Differentiation (SDD) [9], which approximates $\partial_x^n \partial_t^m u(i,j)$ for any integers $m, n \ge 0$ by

$$(S_x D_x)^n (S_t D_t)^m S_x S_t U(i,j)$$

where S_x and S_t are 1-D smoothing operators along space and time respectively, D_x and D_t are numerical differentiation operators with respect to space and time respectively, and $(\cdot)^m$ means applying the operator repeatedly for m times.

In this paper, we assume that the grid is uniform with step size $\Delta x > 0$ in space and $\Delta t > 0$ in time. We use 5-point-central difference for both D_x and D_t, that is

$$D_x U(x_i, t_n) = \frac{-U(x_{i+2}, t_n) + 8U(x_{i+1}, t_n) - 8U(x_{i-1}, t_n) + U(x_{i-2}, t_n)}{12\Delta x}$$

and similarly for D_t. Here periodic boundary condition is applied in space, and Neumann boundary condition is applied in time. To reduce the influence of the approximation errors near boundary, we only use the interior data for feature construction. As for the smoothing operator in time and space, we use the Savitzy-Golay filter [41], which is a convolution version of the local polynomial fitting. For example, when the boundary condition is periodic, the spatial smoothing operator with the Savitzy-Golay filter is

$$S_x U(i,n) = \sum_{l=\frac{1-w}{2}}^{\frac{w-1}{2}} W_l U(i+l, n), \tag{24}$$

where the integer $w \ge 1$ is the window size, the convolution weights W_l are derived by fitting local data using degree q polynomials for some integer $0 \le q < w$, and they are tabulated in [41]. This filter is available, e.g., using `savgol_filter` from the `scipy` package in Python. In the following numerical section, we use the notation such as SDD-15 to represent using SDD with a window width $w = 15$ in (24). In this paper, we find that more accurate coefficient reconstruction is obtained if ∂_t is approximated by $D_t S_t$ without the second smoothing, thus we modify SDD as such in our experiments.

5. Numerical experiments

For all experiments, we consider general dictionaries parameterized by two positive integers: the highest order of partial derivative, and the maximal number of multiplication of terms. For example, a dictionary that contains partial derivatives of u up to the first order and the products of no more than 2 terms consists of the following features $1, u, u_x, u^2, uu_x$, and u_x^2. Our default dictionary contains 56 terms including all partial derivatives of u up to order 4 and the products of no more than of 3 features. Our experiments in subsubsection 5.3.3 and Table 5 are performed on larger dictionaries to show the stability against dictionary size.

Table 2
A list of PDEs tested in Section 5.

PDE	Model
Advection diffusion equation	$u_t = \partial_x(a(x)u) + bu_{xx}$
Fisher's equation	$u_t = bu_{xx} + a(t)u(1-u)$
Viscous Burgers equation	$u_t = a(x,t)uu_x + b(t)u_{xx}$
Korteweg–De Vries (KdV) equation	$u_t = a(x,t)uu_x + b(x,t)u_{xx}$
Kuramoto–Sivashinsky (KS) equation	$u_t = a(x)uu_x + b(x,t)u_{xx} + c(x,t)u_{xxxx}$
Schrödinger equation	$iu_t = bu_{xx} + a(x,t)u$
Nonlinear Schrödinger (NLS) equation	$iu_t = bu_{xx} + a(x,t)\lvert u\rvert^2 u$

For hyper-parameters, we fix $K_{\max} = 15$, $\rho = 0.015$, and $L = 5$ in all experiments, since we find that GP-IDENT is not sensitive to the choice of $L > 1$ and ρ. We illustrate the effect of RR and ρ with an example in Appendix B. We consider data with $p\%$ Gaussian noise. The noisy data takes the form $U(x_i,t_n) = u(x_i,t_n) + \varepsilon_{i,n}$ for $i = 1,\dots,I$, $n = 1,\dots,N$, with Gaussian noise $\varepsilon_{i,n} \sim \mathcal{N}(0,\sigma^2)$, $i = 1,\dots,I$, $n = 1,\dots,N$, where

$$\sigma = p\% \times \mathrm{std}(\{u(x_i,t_n) \mid i = 1,\dots,I, n = 1,\dots,N\}). \tag{25}$$

Here std$(\cdot)$ stands for the standard deviation of a collection of data.

For the PDE examples, we generate the solution data by the spectral method analogous to [39]. The equation is discretized in space, where the partial derivatives are computed using Fast Fourier Transform (FFT), then the solution is obtained by integrating in time using LSODA [13]. As for examples of PDE systems, i.e., Schrödinger and Nonlinear Schrödinger equations, we generate the data by implicit-explicit finite difference methods where second order partial derivatives are treated implicitly, and the zero-th order terms are treated explicitly. To represent the change in the coefficients, for some fixed time $T_{\max} > 0$, we define

$$\tau_{\pm}(t;s,t_b) = \frac{1}{2} + \frac{1}{2}\tanh\left(\pm\frac{s(t-t_b)}{T_{\max}}\right),\ t \in [0,T_{\max}] \tag{26}$$

to reflect a smooth transition with rate s between different states separated by the break point t_b.

We present numerical experiments to justify the effectiveness of GP-IDENT and compare it with the state-of-the-art identification methods for PDEs with varying coefficients: GLASSO [52], SGTR [39], and rSGTR [24].[1] In [24], DLrSR was proposed to handle sparse noise added to the measurements in a linear system. We note that the selection criteria for GLASSO, SGTR, and rSGTR are based on minimum AIC value, and they are essentially parameter-free. We also compare GP-IDENT with BSP-IDENT, where GPSP is replaced by BSP [18] in GP-IDENT. To show the effectiveness of GP-IDENT, we test it on various types of equations [40,39,24,9] listed in Table 2.

To evaluate the reconstruction accuracy, we calculate the discrete relative L_1-error to measure the coefficient error:

$$e(C_g) = \frac{\sum_{i=1}^{I}\sum_{n=1}^{N}\left|\widehat{C}_g(x_i,t_n) - C_g(x_i,t_n)\right|}{\sum_{i=1}^{I}\sum_{n=1}^{N}\left|C_g(x_i,t_n)\right|} \times 100\% \tag{27}$$

where $\widehat{C}_g$ is the reconstruction of C_g. To quantify the coefficient support identification accuracy, we use the Jaccard index [15] defined as

$$J(\widehat{T},T^*) = \frac{|\widehat{T}\cap T^*|}{|\widehat{T}\cup T^*|}, \tag{28}$$

where $\widehat{T}$ denotes the group index set in the identified model, T^* is the group index set in the true equation, and $|\cdot|$ gives the number of elements in the set. Note that $J(\widehat{T},T^*) = 1$ if and only if $\widehat{T} = T^*$, i.e., the underlying model is exactly identified.

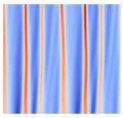
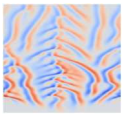
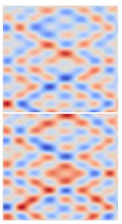
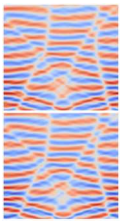
For all the compared methods, the codes are available in `Python`. For BSP-IDENT and GP-IDENT, we implement the generation of feature system in `Python`, and the BSP as well as GPSP algorithms and the identification process in `C++`. We run the experiments using a laptop with 12th Gen Intel(R) Core (TM) i7-12800H, 2400 Mhz, and 14 cores.

5.1. GP-IDENT results on PDEs with space and time varying coefficients

We experiment on several PDEs with space and time dependent coefficients, including the KdV equation, KS equation, Schrödinger equation (Sch), and Nonlinear Schrödinger (NLS) equation. We note that Sch and NLS equations can be regarded as PDE systems for the real and imaginary components of a complex system. For the KdV and KS equations, we use the default dictionary containing 56 terms. As for the PDE systems (Sch and NLS), we use the dictionary containing linear features of partial derivatives of the real and imaginary components up to order 3, and the products up to 3 terms, leading to a total of 165 features. Table 3 shows the trajectories, equations, the coefficient reconstruction errors (27) with clean and noisy data. We present the details of these experiment settings

[1] For GLASSO [52] and SGTR [39], we used the code available at https://github.com/snagcliffs/parametric-discovery; and for rSGTR, https://github.com/junli2019/Robust-Discovery-of-PDEs.

Table 3
GP-IDENT results for equations and systems with space and time varying coefficients. The first column shows the solution trajectory for each equation. For the Schrödinger and NLS equations, the real and imaginary components of u are plotted, respectively. The second column shows the true equations, and GP-IDENT finds the correct features, i.e., the identified equations match the true equations. Once the feature terms are identified, least-square is used to recover the coefficient values. In the third column, we report the relative L_1 errors (27) for the recovered coefficients. For the noisy case (1% noise), we conduct 10 independent experiments and record the mean and standard deviation of the errors.

Trajectory	Equation	Coef. error no noise, 1% noise
KdV 	$u_t = a(x,t)uu_x + b(x,t)u_{xxx}$	u_{xxx} : 4.09%, $20.49 \pm 0.16\%$ uu_x : 0.54%, $20.37 \pm 0.18\%$
KS 	$u_t = a(x)uu_x + b(x,t)u_{xx} + c(x,t)\partial_x^4 u$	u_{xx} : 2.03%, $19.21 \pm 0.31\%$[a] $\partial_x^4 u$: 2.12%, $18.92 \pm 0.30\%$[a] uu_x : 1.05%, $25.61 \pm 0.21\%$[a]
Sch 	$iu_t = 0.5u_{xx} + a(x,t)u$	v : 3.80%, $5.34 \pm 0.12\%$ w : 3.93%, $4.14 \pm 0.11\%$ v_{xx} : 0.78%, $0.87 \pm 0.03\%$ w_{xx} : 0.81%, $0.73 \pm 0.03\%$ $v = \mathrm{Re}(u), w = \mathrm{Im}(u)$
NLS 	$iu_t = -0.5u_{xx} + a(x,t)\lvert u\rvert^2 u$	v_{xx} : 1.74%, $21.48 \pm 0.05\%$ w_{xx} : 1.97%, $21.16 \pm 0.04\%$ v^3 : 0.39%, $2.67 \pm 0.01\%$ $v^2 w$: 0.40%, $2.76 \pm 0.04\%$ vw^2 : 0.50%, $2.61 \pm 0.02\%$ w^3 : 0.36%, $2.44 \pm 0.01\%$ $v = \mathrm{Re}(u), w = \mathrm{Im}(u)$

[a] To identify KS equation from noisy data, we used $\rho = 0.05$ for model selection.

including the coefficients, grid, number of bases, and window size for SDD in Appendix A Table A.8. We note that for the KS equation, a different threshold $\rho = 0.05$ is used.

5.2. Viscous Burgers' equation with space-time dependent coefficients

Consider the following viscous Burgers' equation

$$u_t(x,t) = a(x,t)u(x,t)u_x(x,t) + b(t)u_{xx}(x,t)\,,\ x \in [-2,2), t \in (0,0.02] \tag{29}$$

with the initial condition

$$u(x,0) = \sin(\pi(2x-0.1)) + \cos(\pi(5x-0.2)) + \cos(\pi(3x-0.3))\cos(\pi(x+0.1)) + \sin(\pi(4x+0.5)) + 5 \tag{30}$$

and space-time dependent coefficients

$$a(x,t) = 4\left(1 + \tau_+\left(t, 10, \frac{0.02}{3}\right)\right)(2 + \sin(\pi x))\,,\ b(t) = 0.8\left(1 + \tau_-\left(t, 10, \frac{0.02}{2}\right)\right), \tag{31}$$

We numerically solve (29) on a 256×256 grid. Fig. 4 (a), (b), and (c) show the trajectory data, the true coefficient for u_{xx}, and that for uu_x, respectively.

5.2.1. GP-IDENT result

We use 4 bases in space and 7 bases in time to approximate the coefficients. Fig. 4 (d) shows the absolute error of the trajectory simulated by the identified PDE by GP-IDENT, which is close to the true trajectory. Fig. 4 (e) and (f), display the relative L_1 errors (%) for the reconstructed coefficients of u_{xx} and uu_x, respectively. These figures demonstrate an accurate coefficient recovery of (29). In (g)-(i), we show the absolute error of the simulated trajectory and the relative L_1 errors (%) of the reconstructed coefficients

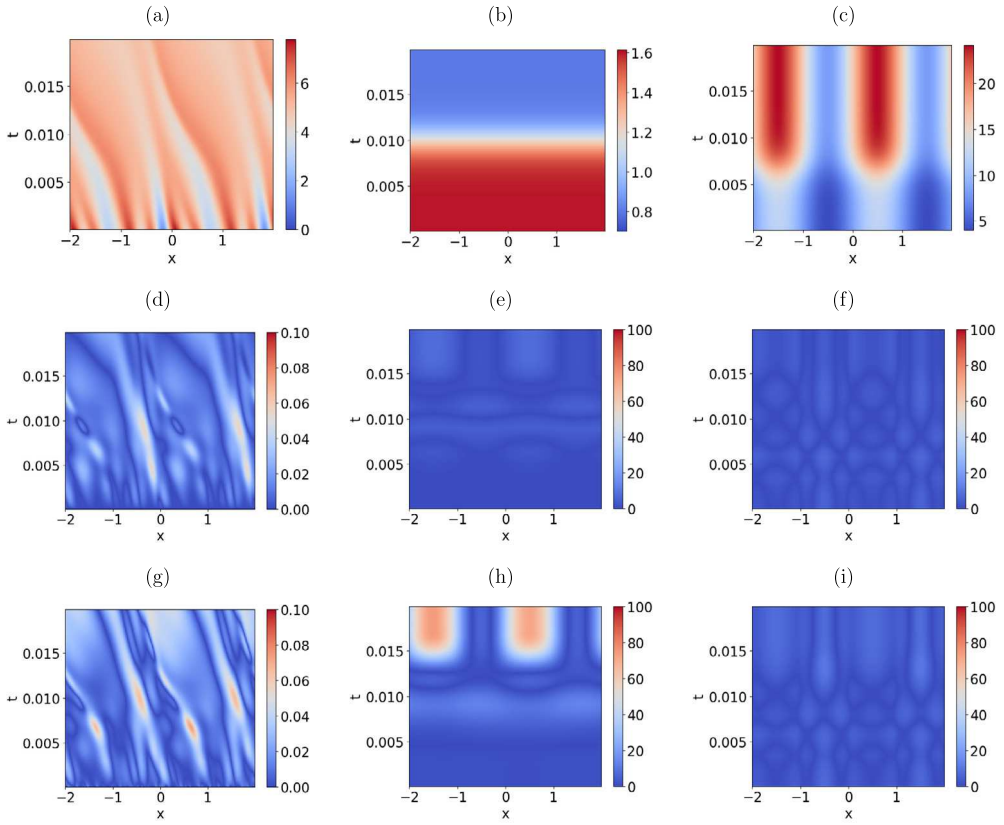


Fig. 4. GP-IDENT result on viscous Burgers equation (29). (a) The true clean trajectory, (b) the true coefficient for u_{xx}, and (c) the true coefficient for uu_x. The second row shows GP-IDENT for clean data: (d) the absolute error of simulation from the identified model, (e) the relative L_1 coefficient error in percentage of the coefficient for u_{xx}, and (f) the relative L_1 coefficient error in percentage for uu_x. The third row shows GP-IDENT for the data with 2% noise: (g) the absolute error of simulation from the identified model, (h) the relative L_1 coefficient error in percentage of the coefficient for u_{xx}, and (i) the relative L_1-error in percentage of the coefficient for uu_x.

when the given data have 2% noise and SDD-9 is used for denoising. GP-IDENT successfully identified the underlying PDE, and the simulated trajectory remains close to the true one. We note that the reconstructed coefficient for u_{xx} deviates from the true ones when $t \in (0.015, 0.02)$ because the observed trajectory in (a) is mostly flat in this region. The flatness (derivatives being close to zero) causes a lack of local dynamics and leads to numerical instability. The coefficient identification on this region is ill-posed.

5.2.2. *Robustness against various level of noise*

We demonstrate the robustness of GP-IDENT and compare with SGTR and BSP-IDENT for various noise levels. Fig. 5 (a) shows the relative L_1 coefficient error for uu_x and u_{xx}, and the relative L_1 error between simulated trajectory using the model identified by GP-IDENT and the true trajectory (green). The coefficient identification for uu_x is robust to noise, yet the coefficient reconstruction for u_{xx} is an ill-posed problem, since the dynamics are flat at some regions, as shown in Fig. 4. Despite that the coefficient error for u_{xx} is relatively large, the simulated trajectory matches the PDE solution with less than 1% error. The reduction of the error in the early stage is caused by over-smoothing of SDD-9 when the noise level is very low.

Fig. 5 (b) shows the Jaccard index between the exact support and the recovered one by SGTR, BSP-IDENT and GP-IDENT. Both SGTR and BSP-IDENT fail to identify the correct features, while GP-IDENT successfully finds the correct model when the noise is below 4%. We note that for SGTR, the optimal model is selected by the minimum of the loss function based on AIC.

5.3. *Advection-diffusion equation with space-dependent coefficients*

Consider the following advection-diffusion equation [39] with spatially dependent coefficients, for $x \in [-5, 5)$, and $t \in (0, 5]$,

$$u_t(x,t) = \partial_x(a(x)u) + 0.1u_{xx} = \partial_x a(x)u + a(x)u_x + 0.1u_{xx} \tag{32}$$

with initial condition $u(x,0) = \cos(2\pi x/5)$, and $a(x) = -1.5 + \cos(2\pi x/5)$. This PDE is solved over a 256×256 (space $\times$ time) grid. When the given data are noisy, SDD plays a critical role. We show in Appendix A Fig. A.9 that noise is significantly amplified in the finite difference scheme; whereas SDD effectively suppresses the perturbation in partial derivatives, thus it helps to identify the true dynamics.

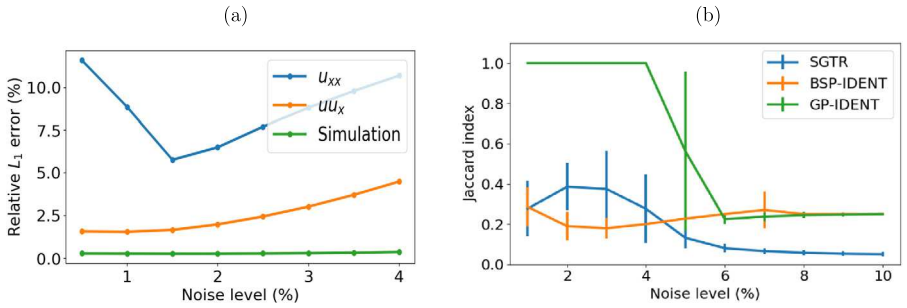


Fig. 5. Varying noise level comparison for the viscous Burgers equation (29): (a) Relative L_1 error for the coefficient for u_{xx} (blue) and uu_x (orange). Green curve shows relative L_1 error between the true trajectory and the simulated trajectory of the identified model by GP-IDENT at various noise levels. Although the coefficient error for u_{xx} is large due to ill-posedness, the simulated trajectory matches with less than 1% error. (b) Accuracy of support identification measured by Jaccard index under various levels of noise. For each noise level, we ran 20 independent experiments using the default dictionary with 56 terms. For BSP-IDENT and GP-IDENT, we used SDD-9 for all levels of noise. GP-IDENT successfully finds the correct model when the noise is below 4%.

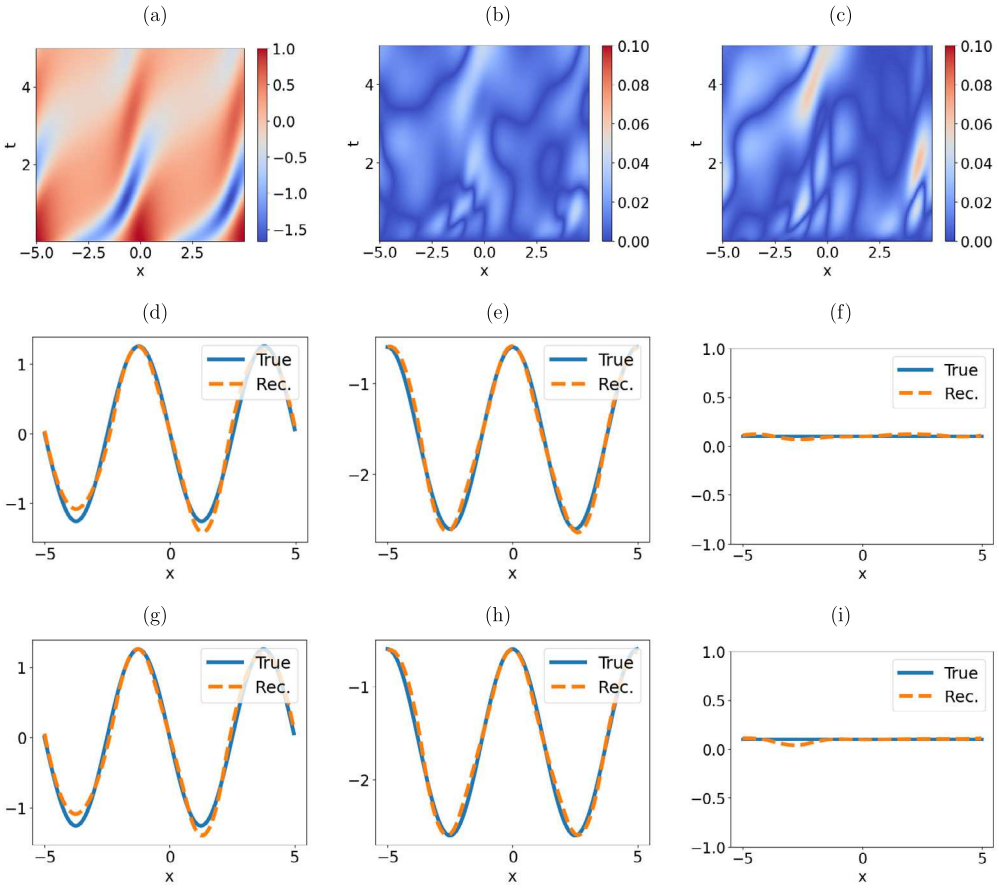


Fig. 6. GP-IDENT result for the advection-diffusion equation (32): (a) observed clean trajectory. (b) and the second row (d)-(f) shows results from the clean data, and (c) and the third row (g)-(i) shows results from the given data with 1% noise (SDD-15 is applied for denoising). The first row shows absolute difference between the true (a) and the trajectory simulated by GP-IDENT. (d) and (g) are reconstruction of the coefficient of u, (e) and (h) of u_x, (f) and (i) of u_{xx}.

5.3.1. GP-IDENT result

For this experiment, we assume that we a priori know that coefficients only vary in space. Using 7 bases in space for the coefficient approximation, GP-IDENT successfully identified the equation (32). Fig. 6 (b) and the second row, (d)-(f) show the reconstruction results with clean data, which stay close to the true coefficient values. We also test GP-IDENT when the data has 1% noise using SDD-15 (Section 4.2). (c) shows the absolute error of the simulated trajectory, and the third row, (g)-(i) show the reconstructed coefficients. GP-IDENT yields robust recovery.

Table 4
Applying GP-IDENT using three different types of bases: only varying in space (S), only in time (T) and both (ST) to the space varying coefficient advection-diffusion equation (32) with and without noise. We present the identified features and sum of squared residual (SSR). For both clean and noisy data, correct features are found with the minimal SSR for varying only in space (S), which is consistent with (32).

	Only in space (S)	Only in time (T)	Both (ST)
	No noise		
features	$\boldsymbol{u, u_x, u_{xx}}$	u_x, u^2, uu_x, u^2u_x, u_x^3	1, u, u_x
SSR	$\mathbf{1.33 \times 10^{-3}}$	9.58×10^{-2}	1.63×10^{-2}
	1% noise		
features	$\boldsymbol{u, u_x, u_{xx}}$	u_x, u^2, uu_x, u^2u_x, u_x^3	1, u, u_x
SSR	$\mathbf{6.76 \times 10^{-3}}$	9.95×10^{-2}	2.13×10^{-2}

Table 5
The advection-diffusion equation (32) identification comparisons: GLASSO [52], SGTR [39], rSGTR [24], BSP-IDENT, and GP-IDENT with three dictionaries and various noise levels. For BSP-IDENT and GP-IDENT, SDD-15 is applied for denoising. Dictionary I has 35 features, II has 56 features, and III has 330 features. Correct support identifications are marked in bold. GLASSO does not converge when Dictionary III is used. GP-IDENT consistently identifies the correct terms.

	No noise				
Method	GLASSO	SGTR	rSGTR	BSP-IDENT	GP-IDENT
Dict. I	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$
Dict. II	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$
Dict. III	–	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$
	1% noise				
Method	GLASSO	SGTR	rSGTR	BSP-IDENT	GP-IDENT
Dict. I	4 terms	u, u_x	u, u_x	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$
Dict. II	$\boldsymbol{u, u_x, u_{xx}}$	u, u_x	u, u_x	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$
Dict. III	–	u, u_x	u, u_x	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$
	3% noise				
Method	GLASSO	SGTR	rSGT	BSP-IDENT	GP-IDENT
Dict. I	$u, u_x, \partial_x^3 u$	5 terms	5 terms	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$
Dict. II	$u, u_x, \partial_x^3 u$	5 terms	5 terms	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$
Dict. III	–	5 terms	5 terms	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$
	6% noise				
Method	GLASSO	SGTR	rSGTR	BSP-IDENT	GP-IDENT
Dict. I	$u, u_x, \partial_x^3 u$	18 terms	18 terms	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$
Dict. II	$u, u_x, \partial_x^3 u$	10 terms	10 terms	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$
Dict. III	–	6 terms	6 terms	$\boldsymbol{u, u_x, u_{xx}}$	$\boldsymbol{u, u_x, u_{xx}}$

5.3.2. *Space, time or space and time varying coefficients*

In Fig. 6, we assumed that we already know the coefficients depend only on space, and we used the bases only varying in space. Although it is not necessary, the prior knowledge about whether the coefficients of the unknown PDE vary in space, in time, or in both, helps to reduce the complexity and to improve the accuracy of coefficient recovery. In Table 4, we present an experiment, using three different assumptions on the bases, (S) varying only in space, (T) varying only in time, and (ST) space- and time-varying coefficients, and we record the sum of squared residuals (SSR) (11) resulted from these settings. In the first setting (S), we used 7 bases in space; in the second setting (T), we used 5 bases in time; and in the third setting (ST), we took 5 bases in space and 3 bases in time, and used the tensor products of them for the time-space bases. For this example, GP-IDENT identifies the correct three features of (32) in the setting (S), and achieves the minimal SSR, as shown in Table 4. Also, notice that using only in time (T) gives the largest error compared to using both (ST), and it identifies only 1 correct feature with 4 additional wrong ones. This is consistent with the fact that (T) cannot identify spatially varying coefficient values well.

5.3.3. *Stability against dictionary sizes*

We present the results with three dictionaries of different sizes. In Dictionary I, we include partial derivatives of u up to order 3 and their products of no more than 3 terms, in total of 35 features. In Dictionary II, we include partial derivatives of u up to order 4 and their products of no more than 3 terms, in total of 56 features. In Dictionary III, we include partial derivatives of u up to order 6 and their products of no more than 4 terms, in total of 330 features. With each of these dictionaries, we apply GLASSO, SGTR, rSGTR, BSP-IDENT and GP-IDENT to identify (32) from a trajectory of data with or without noise. Table 5 compares the identified features

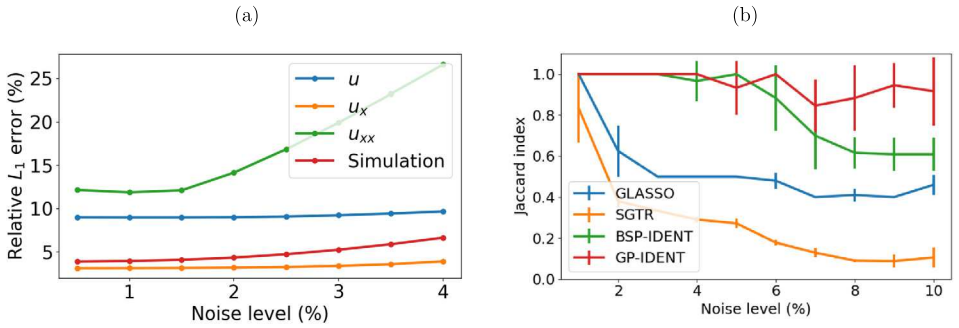


Fig. 7. The advection-diffusion equation (32) identification comparisons with varying noise with Dictionary II. (a) elative L_1 error for the coefficient for u (blue), u_x (orange), and u_{xx} (green). Red curve shows relative L_1 error between the true trajectory and the simulated trajectory of the identified model by GP-IDENT at various noise levels. (b) Jaccard index showing correct support identification with various levels of noise. For BSP-IDENT and GP-IDENT, we used SDD-15 at all levels of noise.

Table 6
Comparison of the computation time for identifying the advection-diffusion equation (32) by GLASSO [52], SGTR [39], rSGTR [24], BSP-IDENT, and GP-IDENT with clean data and different dictionary sizes. GLASSO fails to converge when Dictionary III is used. GP-IDENT and BSP-IDENT show fast convergence.

	Identification time (sec)						
Method	GLASSO	SGTR	rSGTR	BSP-IDENT		GP-IDENT	
$K_{\max}$	–	–	–	10	15	10	15
Dict. I	258.49	7.88	9.45	4.06	7.39	2.71	7.00
Dict. II	356.34	12.77	15.19	2.98	6.98	3.61	9.24
Dict. III	–	145.60	229.46	5.73	9.82	4.19	8.30

of these methods with different noise levels. In this example, GLASSO does not converge when Dictionary III is used, and except for this, all methods have correctly identified the true PDE, when the data has no noise. When the given data have 1% noise, GLASSO identifies the correct features for Dictionary II but not Dictionary I, which suggests that Dictionary II is more co-linear. SGTR fails to identify the correct terms in all cases. We note that in [39], (32) is identified with a smaller dictionary. We find that rSGTR has identical performances as SGTR in terms of feature selection. Both BSP-IDENT and GP-IDENT yield the correct model.

5.3.4. Robustness against noise

We demonstrate the robustness of GP-IDENT and compare with SGTR and BSP-IDENT for various noise levels. Fig. 7 (a) shows the relative L_1 coefficient error for u, u_x, and u_{xx}, and the relative L_1 error between simulated trajectory using the model identified by GP-IDENT and the true trajectory (red). The coefficient identification for u_x is robust to noise. Analogous to the case of Burgers' equation, the coefficient reconstruction for u_{xx} is more challenging. Despite that the coefficient error for u_{xx} is relatively large, the simulated trajectory closely matches the PDE solution.

Fig. 7 (b) shows the Jaccard indices of the identified features by these methods with various noise levels when Dictionary II is used. Overall GP-IDENT and BSP-IDENT outperform the other methods. When the noise level is high, GP-IDENT yields better results than BSP-IDENT.

5.3.5. Computation efficiency

Table 6 shows the computation time for various methods on clean data. For BSP-IDENT and GP-IDENT, the left column of each method records the computation time when $K_{\max} = 10$ and the right column records the time for $K_{\max} = 15$. GP-IDENT and BSP-IDENT show fast convergence. We investigate BSP and GPSP with greater details in Appendix D.

5.4. Fisher's equation with time-dependent coefficients

Consider the Fisher's equation with time-dependent growth rate [35] widely studied in physics and genetics

$$u_t(x,t) = 0.5u_{xx}(x,t) + a(t)u(x,t)(1-u(x,t)), \quad x \in [-5,5), t \in (0,0.8] \tag{33}$$

where

$$a(t) = 1 + \tau_-\left(t; s, \frac{0.8}{3}\right) + \tau_+\left(t; s, \frac{1.6}{3}\right) \tag{34}$$

and $\tau_\pm$ is defined in(26). We take the initial condition

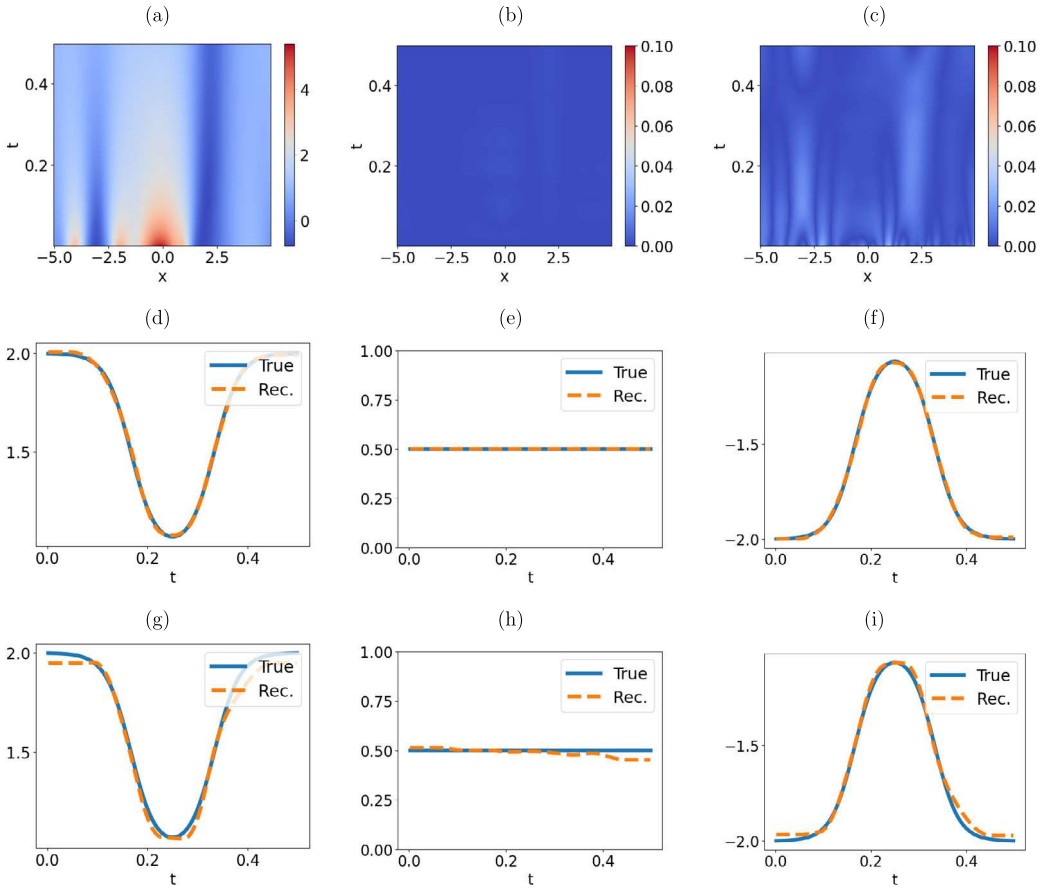


Fig. 8. GP-IDENT result on the Fisher's equation (33): (a) observed clean trajectory. (b) and the second row (d)-(f) shows GP-IDENT results from the clean data, and (c) and the third row (g)-(i) shows results from the given data with 2% noise (SDD-15 is applied for denoising). The first row shows absolute difference between the true (a) and the trajectory simulated by GP-IDENT. (d) and (g) are reconstruction of the coefficient of u, (e) and (h) of u_{xx}, (f) and (i) of u^2.

$$u(x,0) = 5e^{-x^2} + 3e^{-(2x+4)^2} + 2e^{-(3x-3)^2} + 4e^{-(2x+8)^2} + \cos(4(x+1)\pi/10), \tag{35}$$

and numerically solve it on a 256×512 grid for $s = 10$.

5.4.1. GP-IDENT result

For this experiment, we assume that we a priori know that the coefficients are only varying in time. We apply GP-IDENT with 9 bases in time to approximate the coefficients. Fig. 8 (a) shows the clean trajectory, (d)-(f) present the identified coefficients compared to the true coefficients when the given data are clean, and (b) shows the absolute error of the trajectory simulated from the identified model. GP-IDENT identifies varying coefficients accurately. When the data have 2% noise, we apply SDD-15 for denoising, and GP-IDENT identifies the correct model. (c) shows the absolute error of the trajectory simulated from the identified model, and (g)-(i) display the identified coefficients. GP-IDENT is robust to noise.

5.4.2. Comparisons

In Table 7, we compare GLASSO, SGTR, rSGTR, BSP-IDENT, and GP-IDENT for identifying the Fisher's equation (33) with the default dictionary using clean and noisy data. When the given data are clean, all methods identify the correct model. For the data perturbed by noise, GLASSO identifies extra terms, and both SGTR and rSGRT fail to find the correct terms. We note that when the noise level is 1%, all methods identify 3 terms, yet SGTR and rSGTR find wrong terms, and BSP-IDENT and GP-IDENT yield the correct terms. BSP-IDENT and GP-IDENT identify the correct model up to 3% noise. Even with the correct number of terms identified, SGTR and rSGTR may identify wrong features. The performances of both BSP-IDENT and GP-IDENT are more robust.

6. Conclusion

We propose an effective and efficient method, GP-IDENT, for identifying parametric PDEs with space and time-dependent coefficients. Our method generates a few candidates by a greedy algorithm called GPSP at various levels of group sparsity. GPSP algorithm finds a sparse solution to the feature system for any given group sparsity. After generating the candidates, we find the optimal model by considering the smallest sparsity k with a small RR, i.e. $s_k < \rho$. This motivates to find simple equations where RR does not reduce

Table 7
Comparison result for Fisher's equation (33). Table shows identified features of GLASSO, SGTR, rSGTR, BSP-IDENT, and GP-IDENT for data with several levels of noise. For both BSP-IDENT and GP-IDENT, SDD-15 was applied for all levels of noise. Correct identifications are marked in bold. Both BSP-IDENT and GP-IDENT consistently identifies the correct terms.

Method	GLASSO	SGTR	rSGTR	BSP-IDENT	GP-IDENT
No noise	$\boldsymbol{u, u_{xx}, u^2}$	$\boldsymbol{u, u_{xx}, u^2}$	$\boldsymbol{u, u_{xx}, u^2}$	$\boldsymbol{u, u_{xx}, u^2}$	$\boldsymbol{u, u_{xx}, u^2}$
1% noise	4 terms	u_{xx}, u^2, u^3	u_{xx}, u^2, u^3	$\boldsymbol{u, u_{xx}, u^2}$	$\boldsymbol{u, u_{xx}, u^2}$
2% noise	5 terms	6 terms	6 terms	$\boldsymbol{u, u_{xx}, u^2}$	$\boldsymbol{u, u_{xx}, u^2}$
3% noise	5 terms	u^2	u^2	$\boldsymbol{u, u_{xx}, u^2}$	$\boldsymbol{u, u_{xx}, u^2}$

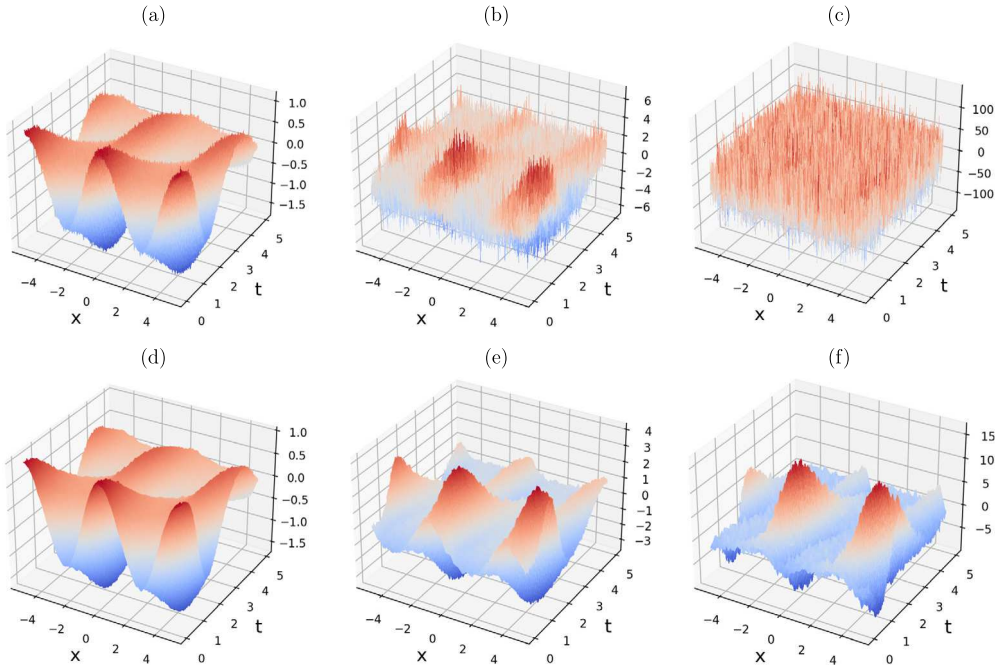


Fig. A.9. For advection-diffusion equation (32), influence of noise and effectiveness of SDD: (a) A noisy trajectory with 10% noise, (b) u_x and (c) u_{xx} computed from the noisy data without denoising. With SDD-15 in the second row, (d) denoised u, (e) denoised u_x, and (f) denoised u_{xx} are more stabilized.

further by adding more complex terms. We demonstrate the effectiveness and efficiency of GP-IDENT on various types of PDEs and compare it with the state-of-the-art methods for PDE identification with varying coefficients. In our experiments, GP-IDENT consistently yields accurate and robust results. To further improve the identification accuracy, especially with high levels of noise, a model selection criterion which is adaptive to noise level may need further investigation.

GP-IDENT can be extended to higher dimensions by expanding the dictionary as well as the basis functions to represent variations in the additional dimensions. For example, in the case of 2D space, if we use 5 bases in both dimensions in space and 5 bases in time, and we consider partial derivatives up to order 4 and their products of no more than 3 terms, the size of the dictionary grows to 816, and the feature system has $816 \times 5 \times 5 \times 5 = 102,000$ columns. When extending to higher dimensions, one may consider local polynomial approximation to reduce the complexity. This method may be applicable to more general PDEs, also to the case of scattered data (not sampled on a uniform grid), which we will investigate in the future.

CRediT authorship contribution statement

Yuchen He: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Sung Ha Kang:** Conceptualization, Methodology, Writing – review & editing. **Wenjing Liao:** Conceptualization, Methodology, Writing – review & editing. **Hao Liu:** Conceptualization, Methodology, Validation, Writing – review & editing. **Yingjie Liu:** Conceptualization, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table A.8
Details of equations tested in Table 3. The Grid column shows the space mesh size × the time mesh size on top, and the space domain × the time domain on the bottom. The Bases column shows the number of bases used for space and time respectively. The SDD column records the smoothing window size of SDD for the noisy data in each case.

Model	Coefficients	Grid	Bases	SDD
KdV	$\begin{cases} a(x) & = 0.5 \cdot (2 + 0.3\cos(\pi x/2)) \cdot \\ & (1 + \tau_+(t; 10, 0.05)) \\ b(x,t) & = 0.01 \cdot (0.5 + 0.1\sin(\pi x/2)) \cdot \\ & (1 + \tau_-(t; 10, 0.05)) \end{cases}$	256×512 $[-2, 2) \times [0, 0.1]$	5, 5	5
KS	$\begin{cases} a(x) & = 2 + \sin(2\pi x/30)/4 \\ b(x,t) & = (-1 + e^{-(x-2)^2/5}/4) \cdot \\ & (2 + \tau_+(t, 5, 30)) \\ c(x,t) & = (-1 - e^{-(x+2)^2/5}/4) \cdot \\ & (2 + \tau_+(t, 5, 30)) \end{cases}$	512×512 $[-30, 30) \times [0, 60]$	9, 5	15
Sch	$a(x,t) = -5\cos(\pi x/2) \cdot (0.5 + \tau_+(t; 5, 0.2))$	100×2000 $[-2, 2) \times [0, 2]$	5, 5	7
NLS	$a(x,t) = (1 + 0.2\cos(\pi x/2)) \cdot (1 + 0.5\tau_+(t; 5, 0.2))$	100×2000 $[-2, 2) \times [0, 0.5]$	5, 5	7

Data availability

No data was used for the research described in the article.

Acknowledgements

This work was supported by Simons Foundation 584960 (S. H. Kang), NSF grant NSF-DMS 2145167 (W. Liao), HKBU 179356 (H. Liu) and NSFC 12201530 (H. Liu).

Appendix A. Effect of SDD, and the experiment settings for Table 3 with space and time varying equations

Fig. A.9 shows that when the given data are noisy, it is significantly amplified in the finite difference scheme. SDD effectively suppresses the perturbation in partial derivatives, and helps to identify the true dynamics.

In Table A.8, we present the details of these experiment settings including the coefficients, grid, number of bases, and window size for SDD for the experiments in Table 3.

Appendix B. Effects of Reduction in Residual (RR)

In this paper, we propose the RR scores (12) to select the identified PDE from a pool of candidates given by GPSP at various levels of sparsity. Using the Burgers' equation (29) as an example, Fig. B.10 demonstrates that the RR scores are effective in selecting the correct model. For both (a) and (b), the black curves are when there is no noise, and the gray curves are for 2% noise. (a) shows residuals for each sparsity level. As the sparsity level gets bigger, the residual curves fluctuate since different sparsity levels are produced by GPSP individually. For example, when the sparsity is 1, the candidate contains u_x. When the sparsity is 2, u_x is removed, and the correct features u_{xx}, uu_x are included. If the sparsity level is 3, the correct feature u_{xx} is not selected, which leads

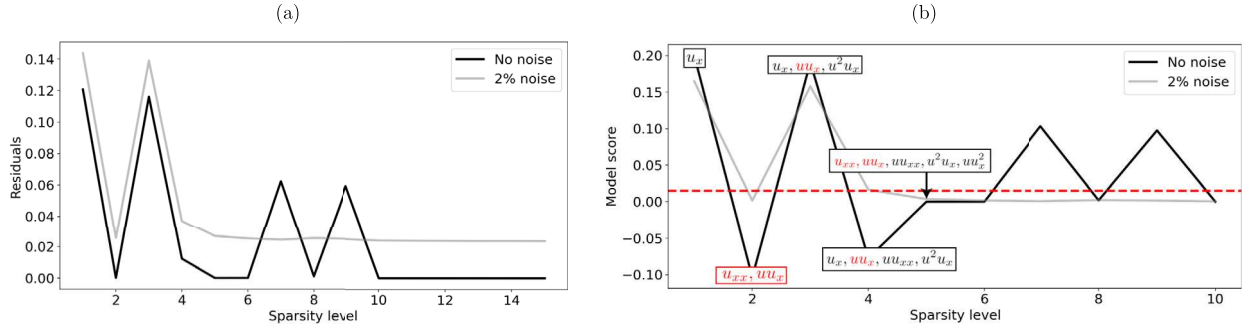


Fig. B.10. Effects of RR for the viscous Burgers' equation (29). For both graphs, the black curve is when there is no noise, and the gray curve is for 2% noise. (a) Residuals of the candidate models from GPSP of various sparsity levels ($K_{\max} = 15$). (b) RR score in (12) for candidates generated by GPSP using $L = 5$. The red dashed curve represents the default threshold $\rho = 0.015$, and the identified model is the one whose score first hits below ρ. The correct features are marked in red in (b).

to an increment of residuals. The residual curves do not give clear indications about the optimal models. In (b), we show the RR curves as well as the threshold $\rho = 0.015$ used in (13) marked by the dashed red line. Note that after sparsity 2, more complex models do not contribute to significant reduction in the residuals. When the given data have noise, the candidates' RR scores become less oscillatory, and the score for the correct model approaches the threshold $\rho = 0.015$. This is commonly observed in other PDEs as well.

Appendix C. Explanation of GPSP over BSP in PDE identification

Consider the transport equation with a constant speed $a \neq 0$

$$u_t(x,t) = au_x(x,t) \tag{C.1}$$

and its solution $f(x+at)$ for some smooth function f, which is nowhere zero. If the hypothesis space contains $f(x+at)$ and the dictionary contains u_x, uu_x, it is possible to confuse (C.1) with

$$u_t(x,t) = \frac{a}{f(x+at)}u(x,t)u_x(x,t) \text{ or } u_t(x,t) = a_1u_x(x,t) + \frac{a_2}{f(x+at)}u(x,t)u_x(x,t) \tag{C.2}$$

where $a_1 + a_2 = a$ and $a_1, a_2 \neq 0$, in which case, all these PDE models are valid.

In practice, the dimension of the hypothesis space M is finite, and the hypothesis space is confined by the resolution of the sampling grid for numerical stability. Ideally, the PDE model with the least coefficient approximation error by the hypothesis space should be selected, and this is where GPSP differs from BSP.

We denote $g(x,t) = \frac{1}{f(x+at)}$ and decompose $g(x,t) = g_M(x,t) + e_M(x,t)$ where g_M is the orthogonal projection of g to $\mathcal{H}_M$, and $e_N \perp \mathcal{H}_M$ denotes the residual. For simplicity, we assume normalization is applied and the dictionary is simply $\{u_x, uu_x\}$. We compare GPSP with BSP when the sparsity level is fixed at 1, that is, each method selects just one feature, and we focus on the selection in the initial step. In BSP, we are comparing

$$\sqrt{\sum_{m=1}^{M}\left(\langle u_t, B_m u_x\rangle\right)^2} \text{ with } \sqrt{\sum_{m=1}^{M}\left(\langle u_t, B_m uu_x\rangle\right)^2} \tag{C.3}$$

where $\{B_m\}_{m=1}^M$ are basis functions and the inner product is understood as operations over the grid points, for example,

$$\langle u_t, B_m u_x\rangle = \sum_{i=1}^{I}\sum_{n=1}^{N} u_t(x_i,t_n)B_m(x_i,t_n)u_x(x_i,t_n) \tag{C.4}$$

By the hypothesis space approximation, (C.1) and (C.2), we have

$$\langle u_t, B_m u_x\rangle = a\langle (g_M + e_M)uu_x, B_m u_x\rangle = a\langle g_M, B_m uu_x^2\rangle + a\langle e_M, B_m uu_x^2\rangle \tag{C.5}$$

$$\langle u_t, B_m uu_x\rangle = a\langle u_x, B_m uu_x\rangle = a\langle g_M, B_m uu_x^2\rangle + a\langle 1 - g_M, B_m uu_x^2\rangle \tag{C.6}$$

thus

$$\langle u_t, B_m uu_x\rangle - \langle u_t, B_m u_x\rangle = a\langle 1 - g, B_m uu_x^2\rangle. \tag{C.7}$$

It indicates that in the first step of BSP, the choice between u_x and uu_x is **independent** of the approximation error e_M; instead, the sign of a as well as the magnitude of the trajectory affects the choice. As for GPSP, we compare

$$\frac{\langle \text{Proj}(u_t, \text{span}_m B_m u_x), u_t\rangle}{\|\langle \text{Proj}(u_t, \text{span}_m B_m u_x)\|_2} = \frac{\langle \text{Proj}(au_x, \text{span}_m B_m u_x), u_t\rangle}{\|\langle \text{Proj}(au_x, \text{span}_m B_m u_x)\|_2} = \|u_t\|_2 \tag{C.8}$$

with

$$\frac{\langle \text{Proj}(u_t, \text{span}_m B_m uu_x), u_t\rangle}{\|\langle \text{Proj}(u_t, \text{span}_m B_m uu_x)\|_2} \leq \|u_t\|_2 \tag{C.9}$$

where $\text{Proj}(u_t, \text{span}_m B_m u_x)$ denotes the projection of u_t to the column space spanned by $\{B_m u_x\}_{m=1}^M$. In (C.8), since $u_x \in \text{span}_m B_m u_x$, the projection of au_x to $\text{span}_m B_m u_x$ is $au_x = u_t$. In (C.9), we used the Cauchy-Schwarz inequality. We note that in (C.9), the equality holds if and only if $e_M = 0$. Therefore, we conclude that GPSP will choose u_x over uu_x if the approximation error for the finite dimensional hypothesis space is non-zero. In other words, the choice of GPSP is **dependent** on the approximation error.

Appendix D. Computation efficiency comparison between BSP and GPSP

In Table 6, we compared the identification time for the advection-diffusion equation in (32) with clean data. When $K_{\max} = 10$, both BSP-IDENT and GP-IDENT are faster than the other methods, and when $K_{\max} = 15$, they require more time as more candidates are generated. We note that typically a single iteration of BSP is faster than a single iteration of GPSP, as BSP only computes vectors' inner products while GPSP involves least-square regressions. In the expanding step, GPSP needs to solve linear systems for vector projections while BSP only computes vector-matrix product. In the shrinking step, GPSP computes vector-matrix product while BSP

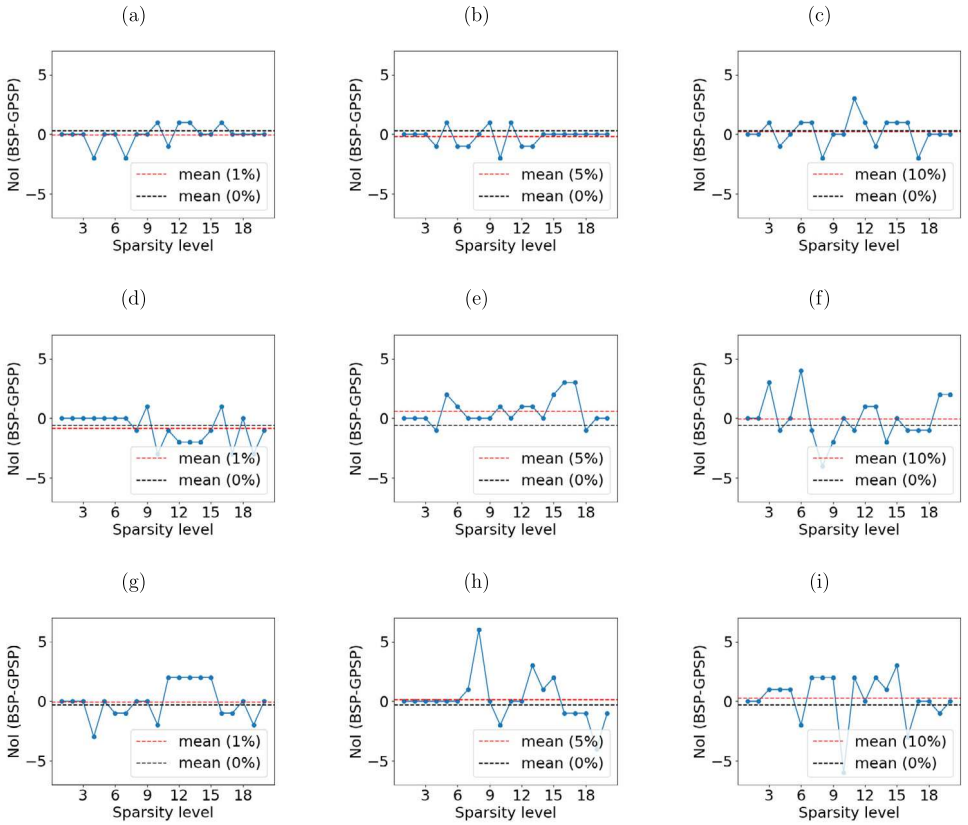


Fig. D.11. For the advection-diffusion equation (32), difference between the number of iterations (NoI) taken till the termination of BSP and GPSP. The blue curve represents the number of iterations of BSP minus that of GPSP. The first row shows using Dictionary I with (a) 1%, (b) 5% and (c) 10% noise. The second row with Dictionary II with (d) 1%, (e) 5% and (f) 10% noise. The third row with Dictionary III with (g) 1%, (h) 5% and (i) 10% noise. Each figure shows the average difference between the number of BSP iterations and the number of GPSP iterations as the sparsity level varies. The dashed black line is the mean of NoI when the data is clean, and the dashed red line is the mean of NoI when there is noise.

only computes L_2 norm of vectors. However, the speed also depends on the number of iterations, the data, the equation, and the dictionary. The total cost of GP-IDENT is comparable to or lower than BP-IDENT. In Fig. D.11, we report the difference of the number of iterations for BSP and GPSP with different noise levels and dictionaries. We observe that in general, GPSP requires fewer iterations than BSP when the noise level is high, and the dictionary size has an effect on this difference.

References

[1] J. Bongard, H. Lipson, Automated reverse engineering of nonlinear dynamical systems, Proc. Natl. Acad. Sci. 104 (24) (2007) 9943–9948.
[2] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. 113 (15) (2016) 3932–3937.
[3] A. Carderera, S. Pokutta, C. Schütte, M. Weiser CINDy, Conditional gradient-based identification of non-linear dynamics-noise-robust recovery, arXiv preprint, arXiv:2101.02630, 2021.
[4] A. Carvalho, J.A. Langa, J. Robinson, Attractors for Infinite-Dimensional Non-autonomous Dynamical Systems, vol. 182, Springer Science & Business Media, 2012.
[5] K. Champion, P. Zheng, A.Y. Aravkin, S.L. Brunton, J.N. Kutz, A unified sparse optimization framework to learn parsimonious physics-informed models from data, IEEE Access 8 (2020) 169259–169271.
[6] W. Dai, O. Milenkovic, Subspace pursuit for compressive sensing signal reconstruction, IEEE Trans. Inf. Theory 55 (5) (2009) 2230–2249.
[7] C. De Boor, B(asic)-spline basics, Technical report, Wisconsin Univ-Madison Mathematics Research Center, 1986.
[8] V.A. Galaktionov, J.L. Vázquez, A Stability Technique for Evolution Partial Differential Equations: a Dynamical Systems Approach, vol. 56, Springer Science & Business Media, 2003.
[9] Y. He, S.-H. Kang, W. Liao, H. Liu, Y. Liu, Robust identification of differential equations by numerical techniques from a single set of noisy observation, SIAM J. Sci. Comput. 44 (3) (2022) A1145–A1175.
[10] Y. He, S.H. Kang, W. Liao, Y. Liu, et al., Numerical identification of nonlocal potentials in aggregation, Commun. Comput. Phys. 32 (3) (2022) 638–670.
[11] Y. He, N. Suh, X. Huo, S.H. Kang, Y. Mei, Asymptotic theory of ℓ_1-regularized PDE identification from a single noisy trajectory, SIAM/ASA J. Uncertain. Quantificat. 10 (3) (2022) 1012–1036.
[12] Y. He, H. Zhao, Y. Zhong, How much can one learn a partial differential equation from its solution?, arXiv preprint, arXiv:2204.04602, 2022.
[13] A.C. Hindmarsh, ODEPACK, a systematized collection of ODE solvers, Sci. Comput. (1983) 55–64.
[14] M. Izadi, J. Abdollahi, S.S. Dubljevic, PDE backstepping control of one-dimensional heat equation with time-varying domain, Automatica 54 (2015) 41–48.
[15] P. Jaccard, The distribution of the flora in the Alpine zone. 1, New Phytol. 11 (2) (1912) 37–50.

[16] E. Kaiser, J.N. Kutz, S.L. Brunton, Sparse identification of nonlinear dynamics for model predictive control in the low-data limit, Proc. R. Soc. A 474 (2219) (2018) 20180335.
[17] B. Kaltenbacher, Identification of nonlinear coefficients in hyperbolic pdes, with application to piezoelectricity, in: Control of Coupled Partial Differential Equations, Springer, 2007, pp. 193–215.
[18] A. Kamali, M.A. Sahaf, A.D. Hooseini, A. Tadaion, Block subspace pursuit for block-sparse signal reconstruction, Iran. J. Sci. Technol. Trans. Electr. Eng. 37 (E1) (2013) 1.
[19] S.H. Kang, W. Liao, Y. Liu Ident, Identifying differential equations with numerical time evolution, J. Sci. Comput. 87 (1) (2021) 1–27.
[20] A.A. Kaptanoglu, J.L. Callaham, A. Aravkin, C.J. Hansen, S.L. Brunton, Promoting global stability in data-driven models of quadratic nonlinear dynamics, Phys. Rev. Fluids 6 (9) (2021) 094401.
[21] S. Kerschbaum, J. Deutscher, Backstepping control of coupled linear parabolic pdes with space and time dependent coefficients, IEEE Trans. Autom. Control 65 (7) (2019) 3060–3067.
[22] Y. Kuramoto, Diffusion-induced chaos in reaction systems, Prog. Theor. Phys. Suppl. 64 (1978) 346–367.
[23] U. Langer, M. Neumüller, A. Schafelner, Space-time finite element methods for parabolic evolution problems with variable coefficients, in: Advanced Finite Element Methods with Applications: Selected Papers from the 30th Chemnitz Finite Element Symposium 2017 30, Springer, 2019, pp. 247–275.
[24] J. Li, G. Sun, G. Zhao, H.L. Li-wei, Robust low-rank discovery of data-driven partial differential equations, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 767–774.
[25] J.-C. Loiseau, S.L. Brunton, Constrained sparse Galerkin regression, J. Fluid Mech. 838 (2018) 42–67.
[26] Z. Long, Y. Lu, B. Dong, Pde-net 2.0: learning pdes from data with a numeric-symbolic hybrid deep network, J. Comput. Phys. 399 (2019) 108925.
[27] Z. Long, Y. Lu, X. Ma, B. Dong PDE-net, Learning PDE from data, in: International Conference on Machine Learning, in: PMLR, 2018, pp. 3208–3216.
[28] A.J. Majda, A.L. Bertozzi, A. Ogawa, Vorticity and incompressible flow, in: Cambridge Texts in Applied Mathematics, Appl. Mech. Rev. 55 (4) (2002) B77–B78.
[29] N.M. Mangan, J.N. Kutz, S.L. Brunton, J.L. Proctor, Model selection for dynamical systems via sparse regression and information criteria, Proc. R. Soc. A, Math. Phys. Eng. Sci. 473 (2204) (2017) 20170009.
[30] D.A. Messenger, D.M. Bortz, Weak SINDy for partial differential equations, J. Comput. Phys. (2021) 110525.
[31] D.A. Messenger, D.M. Bortz, Weak SINDy: Galerkin-based data-driven model selection, Multiscale Model. Simul. 19 (3) (2021) 1474–1497.
[32] T. Meurer, A. Kugi, Trajectory planning for boundary controlled parabolic pdes with varying parameters on higher-dimensional spatial domains, IEEE Trans. Autom. Control 54 (8) (2009) 1854–1868.
[33] A.C. Newell, Solitons in Mathematics and Physics, SIAM, 1985.
[34] J.S. North, C.K. Wikle, E.M. Schliep, A review of data-driven discovery for dynamic systems, arXiv preprint, arXiv:2210.10663, 2022.
[35] A. Öğün, C. Kart, Exact solutions of Fisher and generalized Fisher equations with variable coefficients, Acta Math. Appl. Sin., Engl. Ser. 23 (2007) 563–568.
[36] L. Piegl, W. Tiller, The NURBS Book, Springer Science & Business Media, 1996.
[37] C. Rao, P. Ren, Y. Liu, H. Sun, Discovering nonlinear pdes from scarce data with physics-encoded learning, arXiv preprint, arXiv:2201.12354, 2022.
[38] R. Rudnicki, Chaos for some infinite-dimensional dynamical systems, Math. Methods Appl. Sci. 27 (6) (2004) 723–738.
[39] S.H. Rudy, A. Alla, S.L. Brunton, J.N. Kutz, Data-driven identification of parametric partial differential equations, SIAM J. Appl. Dyn. Syst. 18 (2019) 643–660.
[40] S.H. Rudy, S.L. Brunton, J.L. Proctor, J.N. Kutz, Data-driven discovery of partial differential equations, Sci. Adv. 3 (4) (2017) e1602614.
[41] A. Savitzky, M.J. Golay, Smoothing and differentiation of data by simplified least squares procedures, Anal. Chem. 36 (8) (1964) 1627–1639.
[42] H. Schaeffer, Learning partial differential equations via data discovery and sparse optimization, Proc. R. Soc. A, Math. Phys. Eng. Sci. 473 (2197) (2017) 20160446.
[43] M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data, Science 324 (5923) (2009) 81–85.
[44] I.J. Schoenberg, Contributions to the problem of approximation of equidistant data by analytic functions: Part a.—On the problem of smoothing or graduation. A first class of analytic approximation formulae, in: IJ Schoenberg Selected Papers, 1988, pp. 3–57.
[45] M. Tang, W. Liao, R. Kuske, S.H. Kang. WeakIdent, Weak formulation for identifying differential equations using narrow-fit and trimming, arXiv preprint, arXiv:2211.03134, 2022.
[46] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc., Ser. B, Stat. Methodol. 58 (1) (1996) 267–288.
[47] V. Tikhomirov, A study of the diffusion equation with increase in the amount of substance, and its application to a biological problem, in: Selected Works of AN Kolmogorov, Springer, 1991, pp. 242–270.
[48] H. Vaddireddy, O. San, Equation discovery using fast function extraction: a deterministic symbolic regression approach, Fluids 4 (2) (2019) 111.
[49] R. Vázquez Valenzuela, E. Trélat, J.M. Coron, Control for fast and stable laminar-to-high-Reynolds-numbers transfer in a 2d Navier-Stokes channel flow, Discrete Contin. Dyn. Syst., Ser. B 10 (4) (2008) 925–956.
[50] K. Wu, D. Xiu, Data-driven deep learning of partial differential equations in modal space, J. Comput. Methods Phys. 408 (2020) 109307.
[51] H. Xu, D. Zhang, N. Wang, Deep-learning based discovery of partial differential equations in integral form from sparse and noisy data, J. Comput. Methods Phys. 445 (2021) 110592.
[52] M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, J. R. Stat. Soc., Ser. B, Stat. Methodol. 68 (1) (2006) 49–67.
[53] V.E. Zakharov, S.V. Manakov, On the complete integrability of a nonlinear Schrödinger equation, Theor. Math. Phys. 19 (3) (1974) 551–559.
[54] L. Zhang, H. Schaeffer, On the convergence of the SINDy algorithm, Multiscale Model. Simul. 17 (3) (2019) 948–972.
[55] M.M. Zhang, H. Lam, L. Lin, Robust and parallel Bayesian model selection, Comput. Stat. Data Anal. 127 (2018) 229–247.