Numerical Algebra, Control and Optimization

doi:10.3934/naco.2022040

GROUP SPARSE BAYESIAN LEARNING FOR DATA-DRIVEN DISCOVERY OF EXPLICIT MODEL FORMS WITH MULTIPLE PARAMETRIC DATASETS

Luning Sun[1], Pan Du[1], Hao Sun[2]

and Jian-Xun Wang[*1]

[1]Department of Aerospace and Mechanical Engineering
University of Notre Dame, Notre Dame, IN, USA

[2]Gaoling School of Artificial Intelligence
Renmin University of China, Beijing, China

Abstract. Extracting the explicit governing equations of a dynamic system from limited data has attracted increasing attention in the data-driven modeling community. Compared to black-box learning approaches, the sparse-regression-based learning method enables discovering an analytical model form from data, which is more appealing due to its white-box nature. However, distilling explicit equations from real-world measurements with data uncertainty is challenging, where many existing methods are less robust. Moreover, it is unclear how to efficiently learn a parametric system from multiple data sets with different parameters. This paper presents a group sparse Bayesian learning approaches to uncover the explicit model forms of a parametric dynamical system with estimated uncertainties. A deep neural network is constructed to improve the calculation of derivatives from noisy measurements. Group sparsity is leveraged to enable synchronous learning from a group of parametric datasets governed by the equations with the same functional form but different parameter settings. The proposed approach has been studied over a few linear/nonlinear ODE systems in explicit and implicit settings. In particular, a simplified parametric model of intracranial dynamics was identified from multiple synthetic datasets with different patient-specific parameters. The numerical results demonstrated the effectiveness of the proposed approach and the merit of synchronous learning from multiple datasets in a group sparsifying Bayesian setting.

1. **Introduction.** Dynamical systems are ubiquitous in physical, mathematical, and biological fields. In many cases, the underlying physics behind complex dynamics might not be fully understood, and thus principled models are not available. Thanks to ever-increasing data availability, there is a growing trend in identifying predictive models of a dynamical system from a massive amount of observations, known as *system identification* (SI). Traditional SI methods often use polynomials to construct models that describe the relationship between input and output

2020 *Mathematics Subject Classification.* Primary: 58F15, 58F17; Secondary: 53C35.

Key words and phrases. Equation discovery, Bayesian, dynamical systems, machine learning, sparse regression.

*Corresponding author: Jian-Xun Wang.

The paper is handled by Andreas Mang as the guest editor.

signals, e.g., linear/nonlinear auto-regressive with exogenous variable (ARX) models [22, 26]. However, these models are usually less expressive and limited to low-dimensional state space. Recent advances in deep learning have fueled the rapid development of more capable SI models operating in high-dimensional space. Deep neural networks (DNNs) with various sophisticated network architectures have been designed to learn operators in steady [19, 32, 33, 11, 37] or dynamic [42, 17, 12] scenarios of complex physical phenomena, showing great promise in representation and prediction capabilities. Nonetheless, a major drawback of deep learning models is the lack of interpretability. The network prediction is usually expressed as a prolonged nested function, which is black-box in nature.

Instead of identifying a black-box model, extracting analytical equation forms from data is preferable, which has better interpretability and good potential to advance physics-based modeling. One impressive breakthrough along this route is the sparse identification of nonlinear dynamic (SINDy) algorithm [2, 28], which uses sparse linear regression to uncover parsimonious equation forms of dynamical systems from a high-dimensional nonlinear function space (i.e., basis library) given sufficient observation data. The sparsity was achieved by a sequential threshold ridge regression (STRidge) algorithm [2, 28], which recursively determines the sparse solution subjected to hard thresholds. In the past a few years, the SINDy framework has been further improved in various aspects, and many different variants of SINDy have been proposed for, e.g., dynamics with abrupt changes [27], multi-scale features [20, 3], high-dimensional representation [4], model predictive control [15], and library improvement [7], etc. Compared to deep learning (DL)-based methods, the SINDy family has low training cost, better interpretability, and theoretical convergence [41]. However, a critical bottleneck of the SINDy framework lies in its strong dependence on both data quality and quantity, limiting its applicability to scenarios with incomplete, scarce, and noisy data. This limitation is mainly due to the requirement of derivative information from data, which are usually based on finite difference (FD) methods in vanilla SINDy and its variants. Latest studies show a good potential of combining deep learning techniques and SINDy to handle data sparsity and noise [18, 16, 8, 5, 29]. In particular, DNNs can be used for denoising and derivative computation by fitting the data in a decoupled [29, 38, 39] or coupled manner [18, 16, 8, 5].

Despite the success of these remedies, uncertainties introduced from data/library imperfection and their impacts on the model discovery process cannot be quantified in these deterministic frameworks. As an alternative, people explored the formulation of the equation discovery in a Bayesian setting, known as Sparse Bayesian Learning (SBL). The general idea is to impose sparsity into classic Bayesian inference algorithms (e.g., Bayesian linear regression) by using either sparsity-promoting priors or more *ad hoc* threshold-based methods. For example, Tipping and coworkers proposed sparsity-promoting priors for various classification and regression problems via marginal likelihood maximization methods [34, 35, 1, 9, 10]. Recently, Zhang and Lin [43, 44] used this idea to extend the SINDy to a Bayesian formulation, enabling equation discovery with error bars. Pan et al. [23, 24, 25] leveraged sparsity-promoting priors for the development of a Bayesian system identification algorithm using alternating direction method of multipliers (ADMM) algorithms. Hirsh et al. [13] compared two sparsifying priors, spike-and-slab prior and regularized horseshoe prior, for equation discovery using Bayesian linear regression. Yang

et al. [40] developed a Bayesian differentiable-programming-based SI method beyond Gaussian assumption using the Hamilton Monte Carlo (HMC) method, where a horseshoe prior is adopted to promote the sparsity. In addition to sparsifying priors, sequential threshold methods can be used to impose sparsity based on *ad hoc* thresholds. For example, Zhang and Liu [45] developed a sequential-threshold Bayesian linear regression method to discover parsimonious equation forms by sequentially pruning redundant terms based on user-specified thresholds.

While these works substantiate the potential of sparse learning in Bayesian settings, significant methodology developments are still needed to deal with data scarcity, noise, and uncertainty propagation. This work presents a sparse Bayesian learning approach, where sparsity is imposed based on a sparsifying prior and sequential thresholding. The contribution of the current work is shown as follows. To improve learning efficiency in data-scarce scenarios, we propose to simultaneously use multiple datasets of the systems governed by the same equation form but with varying parameters based on *group sparsity*. Moreover, we propose to incorporate DL-based denoising techniques in a decoupled manner for data preprocessing. To demonstrate the effectiveness of the proposed techniques, we investigated several dynamical systems with a variety of complexity. Note that the main contribution lies in integrating several techniques in an innovative way for system identification. The rest of the paper is organized as follows. Section 2 introduces the key components of the proposed group-sparsity-based Bayesian learning approach, including library-based Bayesian regression, group sparsity, DNN denoising techniques, and uncertainty propagation. Section 3 presents numerical results for data-driven equation discovery of several dynamic systems governed by ordinary differential equations (ODEs) with varying parameters. In particular, intracranial pressure (ICP) dynamic systems are studied to demonstrate the merits of the proposed method. Finally, Section 4 concludes the paper.

2. Methodology.

Let's consider a group of dynamical systems, which are governed by a parametric ODE system in the general form,

$$\frac{d\boldsymbol{x}}{dt} = \mathcal{F}(\boldsymbol{x}; \boldsymbol{\lambda}), \tag{1}$$

where t is the time coordinate, $\boldsymbol{x} = [x_1(t), x_2(t), \cdots, x_d(t)]^T \in \mathbb{R}^d$ represents the state variable, and $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^d$ represents unknown nonlinear functions parameterized by $\boldsymbol{\lambda}$. The systems are observed at discrete times $t = t_1, t_2, \cdots, t_n$, and the set of observed states $\hat{\boldsymbol{x}}$ at the parameter $\boldsymbol{\lambda_k}$ is denoted by $\hat{\boldsymbol{X}}^{(\boldsymbol{\lambda}_k)} = \{\hat{\boldsymbol{x}}(t_1), \hat{\boldsymbol{x}}(t_2), \cdots, \hat{\boldsymbol{x}}(t_n); \boldsymbol{\lambda}_k\}^T \in \mathbb{R}^{n \times d}$. Our goal here is to explicitly discover the analytical forms of $\mathcal{F}(\cdot)$ given a group of datasets of the systems at n_λ different parameters $\boldsymbol{\lambda}_k$,

$$\left\{\hat{\boldsymbol{X}}^{(\boldsymbol{\lambda}_1)}, \hat{\boldsymbol{X}}^{(\boldsymbol{\lambda}_2)}, \cdots, \hat{\boldsymbol{X}}^{(\boldsymbol{\lambda}_{n_\lambda})}\right\}. \tag{2}$$

This scenario widely exists in many system identification applications. The intracranial pressure (ICP) system is one of the examples. The ICP dynamics are driven by complex interactions among cerebral blood flow (CBF), cerebrospinal fluid (CSF), and soft brain tissues. The underlying governing equations behind the ICP dynamics are unknown, while clinically measured CBF, CSF, and ICP signals of many different patients are available. Assuming the model form $\mathcal{F}(\cdot)$ of governing physics is the same for all patients, it is significant to recover the general model form

$\mathcal{F}(\cdot)$ based on a group of patient-specific datasets (i.e., measured dynamic signals at different patient-specific parameters $\boldsymbol{\lambda}$).

2.1. **Sparse system identification in deterministic settings.** Given one dataset $\hat{\boldsymbol{X}}$ at a fixed parameter $\boldsymbol{\lambda}$, the SI problem can be solved by sparse regression techniques based on a predefined library $\boldsymbol{\Phi}(\boldsymbol{x})$ of m basis functions,

$$\boldsymbol{\Phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \cdots, \phi_m(\boldsymbol{x})] \in \mathbb{R}^m, \tag{3}$$

where $\phi_i : \mathbb{R}^d \to \mathbb{R}, i \in [1, m]$ denotes basis functions, which, for instance, can be the polynomial basis and trigonometric functions of the state. Hence, the matrix of library terms of observed states is defined as,

$$\boldsymbol{\Phi}(\hat{\boldsymbol{X}}) = \left[\boldsymbol{\Phi}\big(\hat{\boldsymbol{x}}(t_1)\big)^T, \boldsymbol{\Phi}\big(\hat{\boldsymbol{x}}(t_2)\big)^T, \cdots, \boldsymbol{\Phi}\big(\hat{\boldsymbol{x}}(t_n)\big)^T\right]^T \in \mathbb{R}^{n\times m}. \tag{4}$$

After calculating the time derivatives $\dot{\hat{\boldsymbol{X}}}$ of the data $\hat{\boldsymbol{X}}$, the equation discovery problem can then be formulated as a linear regression problem,

$$\dot{\hat{\boldsymbol{X}}} = \boldsymbol{\Phi}(\hat{\boldsymbol{X}})\boldsymbol{W}, \tag{5}$$

where $\boldsymbol{W} \in \mathbb{R}^{m\times d}$ denotes the matrix of linear coefficients. Based on the principle of Occam's razor, identifying the most parsimonious model form is preferable in practice, which requires the coefficient matrix $\boldsymbol{W}$ to be as sparse as possible. Hence, the eq. 5 can be solved by using regularized least square methods in a deterministic manner,

$$\boldsymbol{W} = \arg\min_{\boldsymbol{W}'} ||\boldsymbol{\Phi}(\hat{\boldsymbol{X}})\boldsymbol{W}' - \dot{\hat{\boldsymbol{X}}}||_{L_2} + \lambda||\boldsymbol{W}'||_{L_\alpha}, \tag{6}$$

where the sparsity can be promoted via the regularizing term defined by different L_α norms, e.g., $L_\alpha = L_1$ corresponding to LASSO, $L_\alpha = L_2$ corresponding to ridge regression, or $L_\alpha = L_0$ corresponding to sequential threshold methods. For example, the sequential threshold least square regression has been proposed as the original SINDy algorithm [2], which has been later improved in many different aspects. In particular, the SINDy based sequential threshold ridge regression (STRidge) algorithm [30] has been demonstrated to be effective and is widely used, and we referred to it as baseline SINDy in this work.

2.2. **Sparse system identification in Bayesian formulations.** As the data $\hat{\boldsymbol{X}}$ always contain measurement noises, Eq. 5 cannot be exactly satisfied and needs to be reformulated as,

$$\dot{\hat{\boldsymbol{X}}} = \boldsymbol{\Phi}(\hat{\boldsymbol{X}})\boldsymbol{W} + \boldsymbol{\epsilon}, \tag{7}$$

where $\boldsymbol{\epsilon}$ represents the process uncertainty, representing errors in $\dot{\hat{\boldsymbol{X}}}$ due to finite difference (FD) approximation and measurement noises. Although the state measurements can be smoothed using a deep learning based denoising method [29], wiggling still exists, which can be amplified in FD-based derivative reconstructions. In this work, the overall process uncertainty is modeled as a zero-mean multivariate Gaussian random variable $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{B}^{-1}\mathbf{I})$, where the covariance matrix is a diagonal matrix with diagonal entries $\beta_k,\ k \in [1, d]$.

For a dynamical system with d state variables, we will solve d Bayesian linear regression problems separately to identity the weight matrix $\boldsymbol{W}$. The data likelihood

for a given coefficient $\boldsymbol{W}_k, k \in [1, d]$ is written as:

$$p(\dot{\hat{\boldsymbol{X}}}_k|\boldsymbol{W}_k, \beta_k) = \prod_{i=1}^{n}(\frac{\beta_k}{2\pi})\exp(-\frac{\beta_k}{2}||\dot{\hat{\boldsymbol{X}}}_k^{(i)} - \boldsymbol{\Phi}(\hat{\boldsymbol{X}}^{(\boldsymbol{i})})\boldsymbol{W}_k||_2). \tag{8}$$

The prior of the weight function is assumed to be zero-mean Gaussian distribution with only diagonal terms in the covariance matrix, as shown in Eq. 9:

$$p(\boldsymbol{W}_k|\boldsymbol{A}_k) = \prod_{j=1}^{m}\mathcal{N}(\boldsymbol{W}_{kj}|0, \alpha_{kj}^{-1}), \tag{9}$$

where $\boldsymbol{A}_k = [\alpha_{k1}, \alpha_{k2}, ..., \alpha_{km}]^T$ is the hyperprior. Each α_{kj} cooresponds to a single $\boldsymbol{W}_{kj}$. This kind of prior has been proved to be able to promote sparsity [34]. To complete the hierarchical Bayesian model, we assume that $\boldsymbol{A}_k$ and β_k follow the log-uniform distribution. Then, the posterior distribution for all the unknown parameters can be decomposed as follows,

$$p(\boldsymbol{W}_k, \boldsymbol{A}_k, \beta_k|\dot{\hat{\boldsymbol{X}}}_k) \propto p(\boldsymbol{W}_k|\dot{\hat{\boldsymbol{X}}}_k, \boldsymbol{A}_k, \beta_k)p(\boldsymbol{A}_k, \beta_k|\dot{\hat{\boldsymbol{X}}}_k), \tag{10}$$

Faul and Tipping [10] proved that given the log-uniform hyperpriors for $\boldsymbol{A}_k$ and β_k, $p(\boldsymbol{A}_k, \beta_k|\dot{\hat{\boldsymbol{X}}}_k)$ can be approximated by Dirac delta function at $(\hat{\boldsymbol{A}}_{\boldsymbol{k}}, \hat{\beta}_k)$. Therefore, the previous equation can be further simplified as Eq. 11:

$$p(\boldsymbol{W}_k, \boldsymbol{A}_k, \beta_k|\dot{\hat{\boldsymbol{X}}}_k) \propto p(\boldsymbol{W}_k|\dot{\hat{\boldsymbol{X}}}_k, \boldsymbol{A}_k, \beta_k)\delta(\hat{\boldsymbol{A}}_{\boldsymbol{k}}\hat{\beta}_k), \tag{11}$$

where $\hat{\boldsymbol{A}}_{\boldsymbol{k}}$ and $\hat{\beta}_k$ can be estimated based on the fast marginal likelihood maximisation algorithm due to the fast convergence [35]. Specifically, $\hat{\boldsymbol{A}}_{\boldsymbol{k}}$ and $\hat{\beta}_k$ are obtained by maximizing the marginal likelihood as,

$$\begin{aligned}
(\hat{\boldsymbol{A}}_{\boldsymbol{k}}, \hat{\beta}_k) &= \arg\max_{\boldsymbol{A}_k, \beta_k}\{p(\dot{\hat{\boldsymbol{X}}}_{\boldsymbol{k}}|\boldsymbol{A}_k, \beta_k)\} \\
&= \arg\max_{\boldsymbol{A}_k, \beta_k}\left\{\int p(\dot{\hat{\boldsymbol{X}}}_k|\boldsymbol{W}_k, \beta_k)p(\boldsymbol{W}_k|\boldsymbol{A}_k)d\boldsymbol{W}_k\right\} \\
&= \arg\max_{\boldsymbol{A}_k, \beta_k}\Big\{(2\pi)^{-n/2}|\beta_k^{-1}\mathbf{I} + \boldsymbol{\Phi}\tilde{\mathbf{A}}_k{}^{-1}\boldsymbol{\Phi}^T|^{-1/2} \\
&\quad \exp\big\{-\frac{1}{2}\dot{\hat{\boldsymbol{X}}}_k{}^T(\beta_k^{-1}\mathbf{I} + \boldsymbol{\Phi}\tilde{\boldsymbol{A}}_k{}^{-1}\boldsymbol{\Phi}^T)^{-1}\dot{\hat{\boldsymbol{X}}}_k\big\}\Big\},
\end{aligned} \tag{12}$$

where $\tilde{\boldsymbol{A}}_k$ is a diagonal matrix with $[\alpha_{k1}, \alpha_{k2}, ..., \alpha_{km}]$ as its main diagonal entries. To optimize Eq. 12, we first define the loss function as the log form,

$$L(\boldsymbol{A}_k) = -\frac{1}{2}\big[n\log 2\pi + |\log\boldsymbol{C}_k| + \dot{\hat{\boldsymbol{X}}}_k{}^T\boldsymbol{C}_k{}^{-1}\dot{\hat{\boldsymbol{X}}}_k\big] \tag{13}$$

where $\boldsymbol{C}_k = |\beta_k^{-1}\mathbf{I} + \boldsymbol{\Phi}\tilde{\mathbf{A}}_{\boldsymbol{k}}{}^{-1}\boldsymbol{\Phi}^T|$. To consider the dependence of the loss function with respect to a single α_{ki}, $\boldsymbol{C}_k$ can be rewritten as:

$$\begin{aligned}
\boldsymbol{C}_k &= \beta_k^{-1}\mathbf{I} + \sum_{m\neq i}\alpha_{km}^{-1}\phi_m\phi_m^T + \alpha_{ki}^{-1}\phi_i\phi_i^T \\
&= \boldsymbol{C}_{k-i} + \alpha_{ki}^{-1}\phi_i\phi_i^T
\end{aligned} \tag{14}$$

where $\boldsymbol{C}_{k-i}$ is $\boldsymbol{C}_k$ calculated from the full basis vectors except the ith term. By using the matrix determinant and inverse formulas, which is listed in detail in [35],

the loss function can be rewritten as Eq. 15:

$$\begin{aligned} L(\boldsymbol{A}_k) &= -\frac{1}{2}\big[n\log(2\pi) + \log|\boldsymbol{C}_{k-i}| + \dot{\hat{\boldsymbol{X}}}_k{}^T \boldsymbol{C}_{k-i}^{-1}\dot{\hat{\boldsymbol{X}}}_k \\ &\quad - \log\alpha_{ki} + \log(\alpha_{ki} + \phi_i^T \boldsymbol{C}_{k-i}^{-1}\phi_i) - \frac{(\phi_i^T \boldsymbol{C}_{k-i}^{-1}\dot{\hat{\boldsymbol{X}}}_k)^2}{\alpha_{ki} + \phi_i^T \boldsymbol{C}_{k-i}^{-1}\phi_i}\big] \\ &= L(\boldsymbol{A}_{k-i}) + \frac{1}{2}\big[\log\alpha_{ki} - \log(\alpha_{ki} + s_{ki}) + \frac{q_{ki}^2}{\alpha_{ki} + s_{ki}}\big]) \\ &= L(\boldsymbol{A}_{k-i}) + l(\alpha_{ki}) \end{aligned} \tag{15}$$

where s_{ki} and q_{ki} are sparsity factors and quality factors, respectively, and they are defined as $s_{ki} = \phi_i^T \boldsymbol{C}_{k-i}^{-1}\phi_i$ and $q_{ki} = \phi_i^T \boldsymbol{C}_{k-i}^{-1}\dot{\hat{\boldsymbol{X}}}_k$. $\boldsymbol{A}_{k-i}$ contains all the terms of $\boldsymbol{A}_k$ except the ith term. Based on the magnitude of q_i and s_i, we can add, re-estimate ,or delete α_{ki} during the iteration , as well as re-estimate β_k (details see [34, 35]).

Given $\hat{\boldsymbol{A}}_{\boldsymbol{k}}, \hat{\beta}_k$, the posterior distribution for the weight can be derived by marginalizing out the hyper-parameters $\hat{\boldsymbol{A}}_{\boldsymbol{k}}$,$\hat{\beta}_k$ as:

$$\begin{aligned} p(\boldsymbol{W}_k|\dot{\hat{\boldsymbol{X}}}_k) &= \iint p(\boldsymbol{W}_k, \boldsymbol{A}_k, \beta_k|\dot{\hat{\boldsymbol{X}}}_k) d\boldsymbol{A}_k d\beta_k \\ &\approx \iint p(\boldsymbol{W}_k|\dot{\hat{\boldsymbol{X}}}_k, \boldsymbol{A}_k, \beta_k)\delta(\hat{\boldsymbol{A}}_{\boldsymbol{k}}, \hat{\beta}_k) d\boldsymbol{A}_k d\beta_k \\ &= p(\boldsymbol{W}_k|\dot{\hat{\boldsymbol{X}}}_k, \hat{\boldsymbol{A}}_{\boldsymbol{k}}, \hat{\beta}_k) \\ &= \mathcal{N}(\boldsymbol{W}_k|\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k), \end{aligned} \tag{16}$$

where $\hat{\boldsymbol{\mu}}_k = \hat{\beta}_k \hat{\boldsymbol{\Sigma}}_k \boldsymbol{\Phi}^{\boldsymbol{T}} \dot{\hat{\boldsymbol{X}}}_k$ and $\hat{\boldsymbol{\Sigma}}_k = \big[\hat{\beta}_k \boldsymbol{\Phi}^{\boldsymbol{T}}\boldsymbol{\Phi} + \hat{\boldsymbol{A}}_k\big]^{-1}$.

2.3. **Group sparsity-based threshold pruning.** The previous sections introduce a general sparse Bayesian regression approach for identifying systems with "parsimonious" equation forms. However, in practice, we found that the identified $\boldsymbol{W}$ is not always "parsimonious". This is especially true when the data noise is considerable, where redundant terms can be identified, though usually with small magnitudes. To remedy this drawback, additional threshold pruning steps can be employed after the sparse Bayesian inference to promote sparsity [43] further. In this work, we propose a group sparsity-based threshold pruning scheme combined with sparse Bayesian regression to handle multiple datasets governed by the same equation with different parameters. The general idea is to iteratively trim off unlikely (small) library terms after every Bayesian inference step in a group-sparsity regression setting [30]. Suppose we have n_λ groups of data corresponding to n_λ parameter sets, the group Bayesian regression can be formulated as,

$$\begin{bmatrix} \dot{\hat{\boldsymbol{X}}}^1 \\ \dot{\hat{\boldsymbol{X}}}^2 \\ \dots \\ \dot{\hat{\boldsymbol{X}}}^{n_\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi} & \dots & \boldsymbol{0} \\ \dots & \dots & \dots & \dots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{\Phi} \end{bmatrix} \begin{bmatrix} \boldsymbol{W}^1 \\ \boldsymbol{W}^2 \\ \dots \\ \boldsymbol{W}^{n_\lambda} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon}^1 \\ \boldsymbol{\epsilon}^2 \\ \dots \\ \boldsymbol{\epsilon}^{n_\lambda} \end{bmatrix} \tag{17}$$

where $[\boldsymbol{W}_1{}^T, \boldsymbol{W}_2^T, ..., \boldsymbol{W}_{n_\lambda}{}^T]^T$ and $[\boldsymbol{\epsilon}^1, \boldsymbol{\epsilon}^2, ..., \boldsymbol{\epsilon}^{n_\lambda}]^T$ are weight vectors and additive process noises for the group of datasets, respectively; $\boldsymbol{\Phi}$ represents the library and should be the same for every dataset. We can perform the group sparse regression

for all the datasets simultaneously by solving Eq. 17. In order to find the most parsimonious equation form that is the same for all datasets, we conduct the threshold pruning in a group setting. Specifically, the mean vectors of the weights for different datasets are stacked into a matrix $\boldsymbol{\Omega}_{m\times n_\lambda} = [\hat{\boldsymbol{\mu}}^1, \hat{\boldsymbol{\mu}}^2, .., \hat{\boldsymbol{\mu}}^{n_\lambda}]$, where each row corresponds to a different basis function in the library and each column corresponds to one dataset in the dataset group G. We will eliminate the rows with coefficients small in magnitudes,

$$\boldsymbol{\Omega}(i,:) = \mathbf{0}, \quad \text{if} \quad ||\boldsymbol{\Omega}(i,:)||_{L_2} \leq GT^2, \tag{18}$$

where T is the user-specified threshold value. After applying the group sparsity, the library $\boldsymbol{\Phi}$ will be updated by dropping out the unlikely basis, where the corresponding row is $\mathbf{0}$ in $\boldsymbol{\Omega}$. The identified results can be improved compared to the sparse Bayesian inference only by iterating the sparse Bayesian inference and group-sparsity threshold pruning.

2.4. **Deep learning denoising method.** We adopt a Runge-Kutta-based neural network (NN) denoising approach [29] to reconstruct the dynamics from noisy measurements and also to estimate data uncertainty, which will be used as the input for the sparse Bayesian regression. The general idea is to decompose the noisy measurements into two parts: signal and noise, which can be estimated by minimizing the reconstruction error. As an unsupervised learning approach, the reconstruction loss over the entire trajectory can be defined as

$$\mathcal{L}(\boldsymbol{\theta}, \hat{\boldsymbol{\epsilon}}, \hat{\boldsymbol{X}}) = \sum_{j=q+1}^{m-q} \sum_{i=-q}^{i=q} \left\| \hat{\boldsymbol{X}}^{(j+i)} - \left(\mathbf{F}_\theta^i (\hat{\boldsymbol{X}}^{(j)} - \hat{\boldsymbol{\epsilon}}^{(j)}) + \hat{\boldsymbol{\epsilon}}^{(j+i)} \right) \right\| \tag{19}$$

where $\mathbf{F}$ represents the Runge-Kutta based NN approximation with trainable parameters θ of the underlying dynamics, $\hat{\boldsymbol{X}}$ represents noisy measurements, and $\hat{\boldsymbol{\epsilon}}$ represents the noise estimation; the superscript i represents indices of local neighborhood steps of the current step j, ranging from $-q$ to q. More details can be found in ref [29].

2.5. **Sequential threshold group sparse Bayesian learning algorithm.** The overall schematics of the Bayesian SI framework are shown in Fig 1. The algorithmic details are given in Algorithm 1 and Algorithm 2, which is extended from the algorithm in [35].

3. **Numerical results.** The proposed sequential threshold sparse Bayesian learning algorithm is demonstrated by discovering several parametric ODE systems based on groups of multiple datasets. It is worth mentioning that an implicit Ursino-Lodi model governing complex intracranial dynamics is identified from a group of synthetic patient-specific data. Here "synthetic patient-specific" means that the datasets are obtained from the same Ursino-Lodi model forms but with different sets of parameters, which mimic the inter-patient variation. We use "patient-specific" to differ from "population-based" data, which is from the model with a set of averaged parameters. Both the identified coefficients and propagated uncertainty are presented.

To demonstrate the merits of sparsity regularization term and group sparsity pruning, which are two important components in the proposed SI method, we compared the model performance with or without them and demonstrated the effectiveness of learning from multiple parametric datasets simultaneously. In particular, we

Algorithm 1: Sequential threshold group sparse Bayesian learning: step 1 parameter estimating.

Result: Mean ($\hat{\boldsymbol{\mu}}^g$) and covaraince ($\hat{\boldsymbol{\Sigma}}^g$) for the relevant coefficients, estimated data noise B^g for the current state, for $g \in [1, G]$

Parameter Estimating Step;

Set group number G and the threshold value T.

for *each subset* $g = 1 : G$ **do**

Estimate the data noise B^g from the deep learning pre-processing result, initialize a single basis vector $\boldsymbol{\phi}_i$ based on the largest projection with targets $\arg\max(|\boldsymbol{\Phi}^T \dot{\hat{\mathbf{X}}}|)$ and the corresponding $\boldsymbol{A}_i^g$, other columns in $\boldsymbol{A}^g$'s are set to infinity. ;

Compute $\hat{\boldsymbol{\mu}}^g$ and $\hat{\boldsymbol{\Sigma}}^g$, and the initial values of s^g and q^g for all basis;

while *not converged* **do**

1. Select a basis vector $\boldsymbol{\phi_i}$ from the whole library $\boldsymbol{\Phi}$ based on the maximum change in the likelihood function ΔL as define in Eq. 15;

2. Compute the relevance variable $\theta_i^g = (q_i^g)^2 - s_i^g$;

if $\theta_i^g > 0$ *and* $\boldsymbol{A}_i^g < \infty$ **then**

re-estimate $\boldsymbol{A}_i^g$ as $(\boldsymbol{A}_i^g)_{new}$;

else if $\theta_i^g > 0$ *and* $\boldsymbol{A}_i^g = \infty$ **then**

add $\boldsymbol{\phi}_i$ to the model with updated $(\boldsymbol{A}_i^g)_{new}$;

else

delete $\boldsymbol{\phi}_i$ from the model and set $(\boldsymbol{A}_i^g)_{new} = \infty$;

3. Recompute and update $\hat{\boldsymbol{\mu}}^g$ and $\hat{\boldsymbol{\Sigma}}^g$;

if *re-estimate* **then**

$\kappa = (\boldsymbol{\Sigma}_{ii} + ((\boldsymbol{A}_i^g)_{new} - \boldsymbol{A}_i^g)^{-1})^{-1}$;

$(\hat{\boldsymbol{\mu}}^g)_{new} = \hat{\boldsymbol{\mu}}^g - \kappa \mu_i^g \hat{\boldsymbol{\Sigma}}_i^g$;

$(\hat{\boldsymbol{\Sigma}}^g)_{new} = \hat{\boldsymbol{\Sigma}}^g - \kappa \hat{\boldsymbol{\Sigma}}_i^g (\hat{\boldsymbol{\Sigma}}_i^g)^T$;

else if *add* **then**

$$(\hat{\boldsymbol{\mu}}^g)_{new} = \begin{bmatrix} \hat{\boldsymbol{\mu}}^g - \hat{\mu}_i^g (B^g)^2 \hat{\boldsymbol{\Sigma}}^g \boldsymbol{\Phi}^T \Phi_i \\ \hat{\mu}_i^g \end{bmatrix};$$

$$(\hat{\boldsymbol{\Sigma}}^g)_{new} = \begin{bmatrix} \hat{\boldsymbol{\Sigma}}^g + (B^g)^2 \hat{\Sigma}_{ii}^g \hat{\boldsymbol{\Sigma}}^g \boldsymbol{\Phi}^T \Phi_i \Phi_i^T \boldsymbol{\Phi} \hat{\boldsymbol{\Sigma}}^g & -(B^g)^2 \hat{\Sigma}_{ii}^g \hat{\boldsymbol{\Sigma}}^g \boldsymbol{\Phi}^T \Phi_i \\ -(B^g)^2 \hat{\Sigma}_{ii}^g (\hat{\boldsymbol{\Sigma}}^g \boldsymbol{\Phi}^T \Phi_i)^T & \hat{\Sigma}_{ii}^g \end{bmatrix}$$

else if *delete* **then**

$(\hat{\mu}^g)_{new} = \hat{\mu}^g - \frac{\hat{\mu}_i^g}{\hat{\Sigma}_{ii}^g} \hat{\boldsymbol{\Sigma}}_i^g$;

$(\hat{\boldsymbol{\Sigma}}^g)_{new} = \hat{\boldsymbol{\Sigma}}^g - \frac{1}{\hat{\Sigma}_{ii}^g} \hat{\boldsymbol{\Sigma}}_i^g (\hat{\boldsymbol{\Sigma}}_i^g)^T$;

4. If converged $|\log(\boldsymbol{A}_i^g)_{new} - \log(\boldsymbol{A}_i^g)|$ is less than a threshold value, break the loop.

end

end

Algorithm 2: Sequential threshold group sparse Bayesian learning: step 2 parameter pruning

Parameter Pruning Step;

Collect the grouped posterior mean value $\boldsymbol{\Omega} = [\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2, ..., \hat{\boldsymbol{\mu}}_G]$;

Initialize $k = 1$, $p_q = m$, and Flag is True;

while *Flag is True* **do**

for $j = 1 : p_q$ **do**

if $||\boldsymbol{\Omega}(j, :)||_{L_2} \leq GT^2$ **then**

$\boldsymbol{\Omega}(j, :) = 0$

end

Find the nonzeros rows in $\boldsymbol{\Omega}$, record the index as $\boldsymbol{I_q}$;

Update $\Phi = \Phi(:, I_q)$ and p_{k+1} = number of elements in I_q, $G = p_{k+1}$;

if $p_{k+1} = p_k$ **then**

Change Flag to False

else

Go back to the **Parameter Estimating Step**

$k = k + 1$;

end

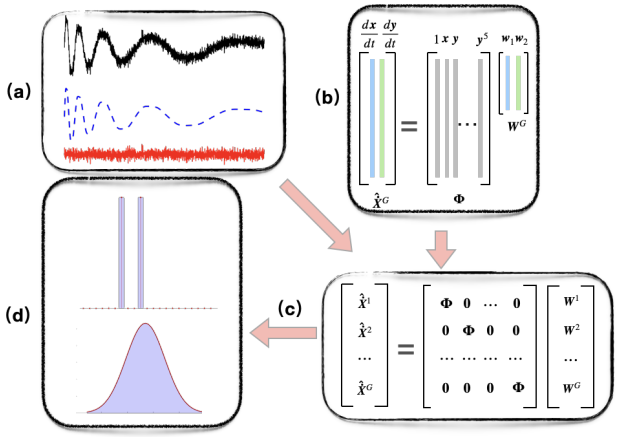

Figure 1. Schematics of the Bayesian group-sparsifying equation discovery framework with (a) neural network pre-processing black: noisy signal. blue: denoised signal. red: estimated noises.; (b) SINDy-type library construction; (c) group sparse Bayesian learning; (d) and example for identified sparse structure and posterior.

consider the following variants: (1) *model with group sparsity* includes both sparsity regularization terms and group-sparsity pruning steps as shown in Algorithms 2.1 and 2.2; (2) *model without group sparsity* only includes the estimation steps using sparsity regularization as shown in Algorithms 2.1; (3) *non-parsimonious model* excludes both regularization terms and group-sparsity pruning. The effectiveness of the group sparsity can be learned by comparing models (1) and (2), while the merits of sparsity regularization can be demonstrated by comparing models (2) and (3).

3.1. **Parametric linear dynamical system.** We first study a linear ODE system parameterized by $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3, \lambda_4]$,

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \lambda_1 & \lambda_2 \\ \lambda_3 & \lambda_4 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}. \tag{20}$$

The library is constructed by the set of polynomials $\{x^p y^q\}$, where $0 \le p+q \le 5$. In the data generating process, we set $\lambda_4 = \lambda_1$ and $\lambda_3 = -\lambda_2$ and nine different sets of parameters are specified ($\lambda_1 \in \{-0.05, -0.10, -0.15\}, \lambda_2 \in \{1.50, 2.00, 2.50\}$). Note that no dependency among different λ_i is assumed *a priori* in the discovering process. The governing equation is simulated with initial conditions $x = 2$ and $y = 0$. And the numerical method is the six-stage, fifth-order, Runge-Kutta method used in ode45 in Matlab. The simulated time is 25s. All the datasets are perturbed with $2\%, 5\%$ and 10% white noises to test the effect of noise. The underlying linear ODE model form can be identified from nine datasets with different parameters using the proposed algorithms. The SI results from the data with 5% noise are summarized in Table 1, where the actual values are marked in blue and redundant terms are marked in red. Note that we present the coefficients as follows: the λ_i denotes the parsimonious coefficients for the equation, while the coefficients of redundant terms are marked by $C(\square)$, where $\square$ denotes the redundant library terms. The model form of this linear system can be precisely discovered; meanwhile, all different sets of parameters are accurately recovered with quantified uncertainties

(standard deviation $\sigma(\lambda)$) for all nine cases. The results without group sparsity are also presented for comparison to demonstrate the merit of using multiple datasets simultaneously based on group sparsity pruning. It is clear that, without group sparsity, redundant constant terms will be identified, making the final result less parsimonious. The results of the data with a larger (10%) noise are presented by Table 12 in the Appendix.Three evaluation metrics, root mean square error (rmse), precision M_P and recall M_R, are presented in Table. 2, which is defined as follows,

$$\begin{aligned} \mathbf{rmse} &= \frac{||\mathbf{C}_{\text{Discovery}} - \mathbf{C}_{\text{True}}||_2}{||\mathbf{C}_{\text{True}}||_2} \\ \mathbf{M_P} &= \frac{||\mathbf{C}_{\text{Discovery}} \odot \mathbf{C}_{\text{True}}||_0}{||\mathbf{C}_{\text{Discovery}}||_0} \\ \mathbf{M_R} &= \frac{||\mathbf{C}_{\text{Discovery}} \odot \mathbf{C}_{\text{True}}||_0}{||\mathbf{C}_{\text{True}}||_0} \end{aligned} \tag{21}$$

where $\mathbf{C}_{\text{Discovery}}$ are the non-zero mean prediction from the posterior distribution and $\mathbf{C}_{\text{True}}$ are the true coefficients of the governing equations. The $\odot$ represents the element-wise product of vectors, and the l_0 norm is the non-zero terms in a vector. We calculated these metrics for three different settings, with group sparsity, without group sparsity, and with a non-parsimonious model. It can be seen that using group sparsity can give better results compared to other settings. To quantify the propagated uncertainty in the state dynamics, we draw 100 Monte Carlo samples from the posterior distribution of the coefficient and forward solve the dynamical system from the initial condition. The propagated dynamics of two selected cases with different parameter sets are shown in Fig. 2, where the uncertainty (3-σ range) is presented as the blue shaded region. And we only show the UQ result for the group sparsity approach. First of all, it can be seen that the dynamics can be very different from each other as the parameters vary. The prediction mean value agrees with the true trajectory very well, and the uncertainty range covers the ground truth but is very tight in this case due to the simplicity of the linear dynamics (see a zoomed-in view in the 2nd row). Only the state variable x is presented here, and the results for y are omitted due to the similarity.

3.2. **Parametric cubic dynamical system.** Following the same convention in Sec. 3.1, we study a parametric cubic dynamical system, which is also parameterized by $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \lambda_3, \lambda_4]$ as given by

$$\frac{d}{dt}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \lambda_1 & \lambda_2 \\ \lambda_3 & \lambda_4 \end{bmatrix}\begin{bmatrix} x^3 \\ y^3 \end{bmatrix}. \tag{22}$$

The parametric datasets are generated by nine different λ sets with the same range as the linear case. The governing equation is simulated with initial conditions $x = 2$ and $y = 0$. The numerical method is the six-stage, fifth-order, Runge-Kutta method used in ode45 in Matlab. The simulated period is 25 s. The library of candidate functions remains the same (i.e., polynomials up to order 5). The identified systems for nine cases from data with 5% noise are summarized in Table 4, where the results with and without group sparsity are listed and compared. The proposed approach can accurately discover the system, which has the exact model forms as the ground truth and the corresponding coefficients are also accurate. In contrast, the results obtained from individual dataset without group sparsity include several redundant terms, e.g., y^2, x^2, x^2y, x^2y^2, and xy^3, making the identified system less parsimonious. This situation deteriorates when data uncertainty increases. When

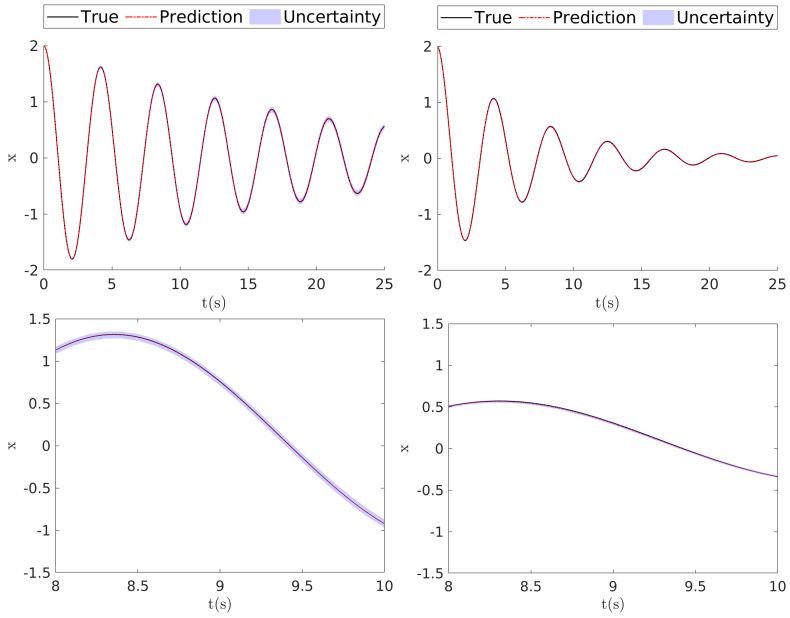

Figure 2. Identified systems of two different set of parameters with 5% noise: (1st column) $\boldsymbol{\lambda} = [-0.05, 1.5, -1.5, -0.05]$ and (2nd column) $\boldsymbol{\lambda} = [-0.15, 2, -2, -0.15]$. Only the trajectory of the state variable x is shown and the propagated uncertainty is given by the 3-σ region. The 2nd row is the zoomed-in view.

data noise is 10%, there are 11 redundant terms identified without group sparsity, and this can be significantly improved by simultaneously using multiple parametric datasets with group sparsity (see Table 13). Table. 3 shows the evaluation metrics for parametric cubic system cases. Group sparsity approach provides the best results in terms of the rmse, M_R and M_P. Fig. 3 shows the propagated dynamics of two selected cases with different parameters, and the uncertainty range is also plotted for state variable $\boldsymbol{x}$. And we only show the UQ result for the group sparsity approach. It can be seen that the dynamics are visually different, and the predicted mean values are close to the true trajectories. Moreover, the predicted uncertainty (3-σ) range covers the ground truth. The multiple peaks of the uncertainty range is due to the uncertainty in the phase difference of the ensemble of forward propagated dynamics. The zoomed-in plot in the second row gives a clearer visualization of the uncertainty range. Again, the dynamics for $\boldsymbol{y}$ are similar and therefore omitted.

3.3. **Parametric Michaelis-Menton Kinetics.** In this section, we will investigate a well-known dynamical system in biochemistry fields, the Michaelis-Menton model, which is a simple yet effective model for enzyme kinetics [21]. It is governed by a single state variable system and can be expressed as Eq. 23,

$$\frac{dx}{dt} = j_x - \frac{V_{max} x}{K_m + x} \tag{23}$$

where j_x is flux source, V_{max} is the maximum reaction rate, and K_m is the concentration of half-maximal reaction rate. Though it appears simple, the Michaelis-Menton system is an implicit system, which means it is not possible to obtain an explicit

Identified systems with group sparsity									
Case	1	2	3	4	5	6	7	8	9
λ_1	-0.0493	-0.0494	-0.0495	-0.1004	-0.1007	-0.0999	-0.1499	-0.1490	-0.1504
$\sigma(\lambda_1)$	0.0019	0.0019	0.0019	0.0017	0.0016	0.0016	0.0015	0.0015	0.0015
λ_2	1.5010	2.0001	2.5008	1.5026	2.0018	2.5009	1.5015	2.0022	2.4996
$\sigma(\lambda_2)$	0.0019	0.0019	0.0019	0.0016	0.0016	0.0016	0.0015	0.0015	0.0015
λ_3	-1.4997	-2.0004	-2.4985	-1.4992	-1.9986	-2.4994	-1.5012	-2.0000	-2.5017
$\sigma(\lambda_3)$	0.0019	0.0019	0.0019	0.0017	0.0016	0.0016	0.0015	0.0015	0.0015
λ_4	-0.0515	-0.0510	-0.0503	-0.1017	-0.1009	-0.1011	-0.1530	-0.1532	-0.1513
$\sigma(\lambda_4)$	0.0019	0.0019	0.0019	0.0016	0.0016	0.0016	0.0015	0.0015	0.0015
Identified systems without group sparsity									
λ_1	-0.0493	-0.0494	-0.0495	-0.1004	-0.1007	-0.0999	-0.1499	-0.1491	-0.1505
$\sigma(\lambda_1)$	0.0019	0.0019	0.0019	0.0017	0.0016	0.0016	0.0015	0.0015	0.0015
λ_2	1.5002	1.9997	2.5007	1.5008	2.0004	2.4998	1.4991	2.0000	2.4978
$\sigma(\lambda_2)$	0.0019	0.0019	0.0019	0.0016	0.0016	0.0016	0.0015	0.0015	0.0015
$C(\text{const1}))$	0	-0.0424	-0.0524	0	0	-0.0370	0	0	0
$\sigma(C(\text{const1}))$	0	0.0017	0.0017	0	0	0.0013	0	0	0
λ_3	-1.4997	-2.0004	-2.4986	-1.4992	-1.9987	-2.4994	-1.5012	-1.9999	-2.5016
$\sigma(\lambda_3)$	0.0019	0.0019	0.0019	0.0017	0.0016	0.0016	0.0015	0.0015	0.0015
λ_4	-0.0507	-0.0506	-0.0502	-0.0996	-0.0993	-0.0999	-0.1499	-0.1507	-0.1493
$\sigma(\lambda_4)$	0.0019	0.0019	0.0019	0.0016	0.0016	0.0016	0.0015	0.0015	0.0015
$C(\text{const2}))$	0	0.0440	0.0551	0	0	0.0411	0	0	0
$\sigma(C(\text{const2})))$	0	0.0017	0.0017	0	0	0.0013	0	0	0
True systems									
λ_1	-0.05	-0.05	-0.05	-0.1	-0.1	-0.1	-0.15	-0.15	-0.15
λ_2	1.5	2.0	2.5	1.5	2.0	2.5	1.5	2.0	2.5
λ_3	-1.5	-2.0	-2.5	-1.5	-2.0	-2.5	-1.5	-2.0	-2.5
λ_4	-0.05	-0.05	-0.05	-0.1	-0.1	-0.1	-0.15	-0.15	-0.15

Table 1. System identification results for the parametric linear systems, $\frac{dx}{dt} = \lambda_1 x + \lambda_2 y$, state $\frac{dy}{dt} = \lambda_3 x + \lambda_4 y$. (5% data noise), where λ_i denotes the relevant coefficient, $C(\text{const1})$ denotes the redundant *constant* coefficient and $\sigma(C(\text{const1}))$ denotes the standard deviation for equation $\frac{dx}{dt} = \lambda_1 x + \lambda_2 y$, while $C(\text{const2})$ and $\sigma(C(\text{const2}))$ are corresponding values for equation $\frac{dy}{dt} = \lambda_3 x + \lambda_4 y$.

Identified systems with group sparsity									
Case	1	2	3	4	5	6	7	8	9
rmse	0.90	0.44	0.51	1.50	0.91	0.44	1.66	1.41	0.62
$\mathbf{M_R}$	1	1	1	1	1	1	1	1	1
$\mathbf{M_P}$	1	1	1	1	1	1	1	1	1
Identified systems without group sparsity									
rmse	0.50	21.59	21.51	0.58	0.58	15.62	0.70	0.41	0.81
$\mathbf{M_R}$	1	0.67	0.67	1	1	0.67	1	1	1
$\mathbf{M_P}$	1	1	1	1	1	1	1	1	1
Identified systems with non-parsimonious model									
rmse	1414.37	1417.92	1425.69	1415.33	1412.46	1417.82	1422.44	1419.52	1411.64
$\mathbf{M_R}$	0.095	0.095	0.095	0.095	0.095	0.095	0.095	0.095	0.095
$\mathbf{M_P}$	1	1	1	1	1	1	1	1	1

Table 2. Different metrics for the parametric linear systems, $\frac{dx}{dt} = \lambda_1 x + \lambda_2 y$, $\frac{dy}{dt} = \lambda_3 x + \lambda_4 y$. (5% data noise)

form like Eq. 1 and construct a library explicitly. To address this issue, we adopted the same idea proposed in the SINDy-PI paper [14] to deal with the rational-form implicit dynamical systems. To be specific, we define the type of dynamical system with the form $g(x, \frac{d\boldsymbol{x}}{dt}) = f(x, \frac{d\boldsymbol{x}}{dt})$ and build two libraries for left- and right- hand side terms, respectively. We can perform a classical explicit discovery for every single term in the left-hand side library. After looping all the possible terms in

Identified systems with group sparsity									
Case	1	2	3	4	5	6	7	8	9
rmse	5.32	2.05	2.86	8.03	6.38	3.97	11.62	8.64	6.14
$\mathbf{M_R}$	1	1	1	1	1	1	1	1	1
$\mathbf{M_P}$	1	1	1	1	1	1	1	1	1
Identified systems without group sparsity									
rmse	61.16	61.25	60.07	48.28	50.58	46.30	40.66	46.12	45.36
$\mathbf{M_R}$	0.67	0.67	0.67	0.67	0.67	0.57	0.67	0.5	0.57
$\mathbf{M_P}$	1	1	1	1	1	1	1	1	1
Identified systems with non-parsimonious model									
rmse	64.10	72.08	70.49	56.96	51.59	55.12	50.09	52.25	46.78
$\mathbf{M_R}$	0.095	0.095	0.095	0.095	0.095	0.095	0.095	0.095	0.095
$\mathbf{M_P}$	1	1	1	1	1	1	1	1	1

Table 3. Different metrics for the parametric cubic systems $\frac{dx}{dt} = \lambda_1 x^3 + \lambda_2 y^3$, $\frac{dy}{dt} = \lambda_3 x^3 + \lambda_4 y^3$ (5% data noise)

Identified systems with group sparsity									
Case	1	2	3	4	5	6	7	8	9
λ_1	-0.0458	-0.0471	-0.0468	-0.0956	-0.0951	-0.0953	-0.1456	-0.1428	-0.1448
$\sigma(\lambda_1)$	0.0111	0.0144	0.0177	0.0073	0.0101	0.0114	0.0054	0.0085	0.0097
λ_2	1.5028	2.0037	2.4908	1.5093	2.0034	2.4931	1.5167	2.0100	2.5086
$\sigma(\lambda_2)$	0.0106	0.0141	0.0177	0.0066	0.0096	0.0112	0.0047	0.0078	0.0094
λ_3	-1.5086	-2.0034	-2.5001	-1.5132	-2.0170	-2.5113	-1.5167	-2.0209	-2.5190
$\sigma(\lambda_3)$	0.0106	0.0139	0.0173	0.0070	0.0104	0.0117	0.0054	0.0074	0.0092
λ_4	-0.0553	-0.0496	-0.0470	-0.1032	-0.1014	-0.0989	-0.1560	-0.1536	-0.1467
$\sigma(\lambda_4)$	0.0101	0.0136	0.0172	0.0064	0.0099	0.0115	0.0046	0.0068	0.0089
Identified systems without group sparsity									
λ_1	-0.0474	-0.0493	-0.0486	-0.0982	-0.1007	-0.1007	-0.1488	-0.1496	-0.1450
$\sigma(\lambda_1)$	0.0021	0.0033	0.0050	0.0016	0.0027	0.0034	0.0012	0.0029	0.0031
λ_2	1.4991	2.0037	2.5034	1.5019	1.9898	2.4886	1.5034	2.0037	2.5127
$\sigma(\lambda_2)$	0.0024	0.0043	0.0068	0.0020	0.0034	0.0045	0.0015	0.0034	0.0038
$C(y^2)$	-0.0921	-0.1260	-0.1530	-0.0737	-0.0966	-0.1098	-0.0603	-0.0849	-0.1174
$\sigma(C(y^2))$	0.0029	0.0050	0.0078	0.0022	0.0038	0.0045	0.0017	0.0037	0.0042
$C(x^2y)$	0	0	0	0	0	0.0405	0	0.415	0
$\sigma(C(x^2y))$	0	0	0	0	0	0.0086	0	0.0068	0
$C(x^2y^2)$	0	0	0	0	0	0	0	0	0.0416
$\sigma(C(x^2y^2))$	0	0	0	0	0	0	0	0	0.0051
$C(x^2y^3)$	0	0	0	0	0	0	0	-0.0381	0
$\sigma(C(x^2y^3))$	0	0	0	0	0	0	0	0.0068	0
λ_3	-1.5033	-1.9984	-2.4967	-1.5000	-2.0028	-2.4993	-1.4975	-2.0010	-2.5003
$\sigma(\lambda_3)$	0.0026	0.0041	0.0054	0.0018	0.0028	0.0044	0.0015	0.0027	0.0030
λ_4	-0.0547	-0.0509	-0.0490	-0.1008	-0.1017	-0.1001	-0.1517	-0.1531	-0.1475
$\sigma(\lambda_4)$	0.0022	0.0035	0.0046	0.0014	0.0023	0.0038	0.0011	0.0022	0.0026
$C(x^2)$	0.0912	0.1189	0.1471	0.0714	0.1052	0.1137	0.0621	0.0819	0.1005
$\sigma(C(x^2))$	0.0026	0.0041	0.0054	0.0018	0.0028	0.0044	0.0015	0.0027	0.0030
True systems									
λ_1	-0.05	-0.05	-0.05	-0.1	-0.1	-0.1	-0.15	-0.15	-0.15
λ_2	1.5	2.0	2.5	1.5	2.0	2.5	1.5	2.0	2.5
λ_3	-1.5	-2.0	-2.5	-1.5	-2.0	-2.5	-1.5	-2.0	-2.5
λ_4	-0.05	-0.05	-0.05	-0.1	-0.1	-0.1	-0.15	-0.15	-0.15

Table 4. System identification results for the parametric cubic systems $\frac{dx}{dt} = \lambda_1 x^3 + \lambda_2 y^3$, state $\frac{dy}{dt} = \lambda_3 x^3 + \lambda_4 y^3$ (5% data noise), where λ_i denotes relevant coefficients, $C(\square)$ denotes the coefficient of redundant terms $\square$.

the left-hand side dictionaries, the best one can be identified by minimizing the prediction error. It is noted that during each single iteration, we need to make sure a single term can not exist in both left and right libraries simultaneously to avoid trivial solutions (e.g., $x\frac{d\boldsymbol{x}}{dt} = x\frac{d\boldsymbol{x}}{dt}$). More details of this process are discussed

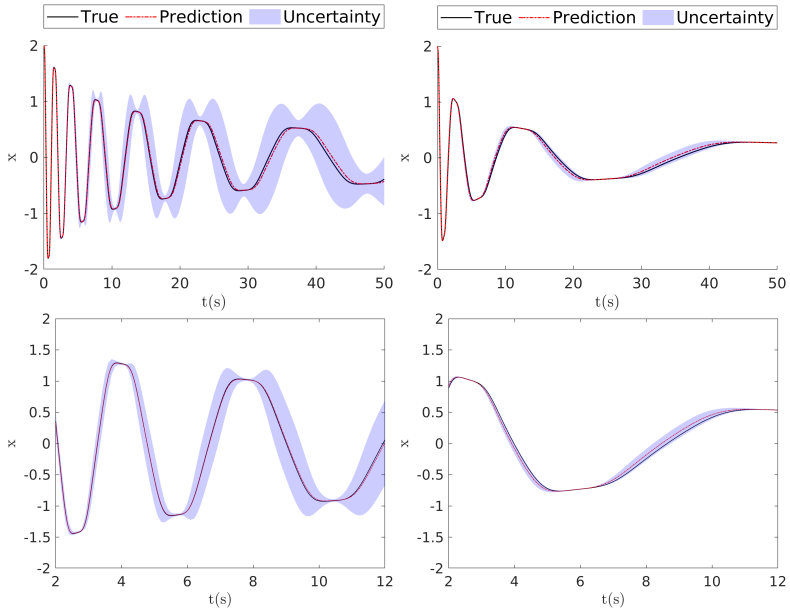

Figure 3. Identified systems of two different sets of parameters with 5% noise: (1st column) $\boldsymbol{\lambda} = [-0.05, 1.5, -1.5, -0.05]$ and (2nd column) $\boldsymbol{\lambda} = [-0.15, 2, -2, -0.15]$. Only the trajectory of the state variable x is shown and the propagated uncertainty is given by the 3-σ region. The 2nd row is the zoomed-in view.

in the SINDy-PI paper [14]. As for the dataset generation, we follow the classical setting in the biochemistry field to vary the initial conditions of x and try to discover the true system from multiple datasets. The initial conditions is chosen as random numbers, and the governing equation is solved by the six-stage, fifth-order, Runge-Kutta method used in ode45 in Matlab. The simulated time span is 10 s.

Ten different parameter pairs of (j_x, V_{max}, K_m) are chosen to generate parametric datasets, which are perturbed with 2% noises. For the implicit system, the iterative SI process proposed in SINDy-PI is less robust for large data noise, which has been reported in the original SINDy-PI work. In the implicit systems, the largest data noise that can be handled is two percent [14]. The system is identified using the proposed group Bayesian approach with the same iterative process to handle the implicit formulation, and the results are summarized in Table 6. In general, the Michaelis-Menton model forms can be discovered, and coefficients for ten different cases are accurately identified. From the study on explicit systems, we found that the merit brought by the group sparsity is less notable when data noise is low. Therefore, there is not much performance difference between the procedures with or without group sparsity pruning. Table. 5 shows the three metrics results. Lastly, the visualization of the propagated parametric system due to the estimated uncertainties is shown in Fig 4. And we only show the UQ result for the group sparsity approach. We can see that the decaying rates vary for different parametric datasets, which can be accurately captured by the predicted mean, and the corresponding uncertainty range reasonably cover the ground truth.

Identified systems with group sparsity										
Case	1	2	3	4	5	6	7	8	9	10
rmse	42.74	9.53	32.39	30.08	12.99	13.40	17.58	12.23	33.10	35.94
$\mathbf{M_R}$	1	1	1	1	1	1	1	1	1	1
$\mathbf{M_P}$	1	1	1	1	1	1	1	1	1	1
Identified systems with non-parsimonious model										
rmse	776.67	704.91	728.43	716.23	625.28	606.92	603.25	748.48	708.18	754.67
$\mathbf{M_R}$	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43
$\mathbf{M_P}$	1	1	1	1	1	1	1	1	1	1

Table 5. Different metrics for parametric MichaelisMenten model (2% data noise)

!htp]

Identified systems with group sparsity										
Term	Coeff	1	2	3	4	5	6	7	8	9	10
dx/dt	λ_1	-0.1608	-0.1926	-0.2607	-0.3621	-0.2077	-0.2870	-0.3776	-0.1872	-0.2591	-0.3541
	$\sigma(\lambda_1)$	0.0288	0.0203	0.226	0.0253	0.0166	0.163	0.0198	0.0199	0.0256	0.0263
x	λ_2	-0.8985	-1.2028	-1.1931	-1.1984	-1.5070	-1.5002	-1.4989	-1.2025	-1.1962	-1.1962
	$\sigma(\lambda_2)$	0.0097	0.0088	0.0096	0.0105	0.0087	0.0085	0.0098	0.0088	0.0107	0.0110
constant	λ_3	0.0645	0.0516	0.0851	0.1154	0.0433	0.0741	0.1043	0.1127	0.1754	0.2354
	$\sigma(\lambda_3)$	0.0154	0.0116	0.0121	0.0127	0.0098	0.0090	0.0105	0.0109	0.0131	0.0127
True systems											
dx/dt	λ_1	-0.2	-0.2	-0.3	-0.4	-0.2	-0.3	-0.4	-0.2	-0.3	-0.4
x	λ_2	-0.9	-1.2	-1.2	-1.2	-1.5	-1.5	-1.5	-1.2	-1.2	-1.2
constant	λ_3	0.06	0.06	0.09	0.12	0.06	0.09	0.12	0.12	0.18	0.24

Table 6. Identification results for parametric MichaelisMenten model (2% data noise), where λ_i denotes the coefficient and $\sigma(\lambda_i)$ denotes standard deviation for the parsimonious library terms.

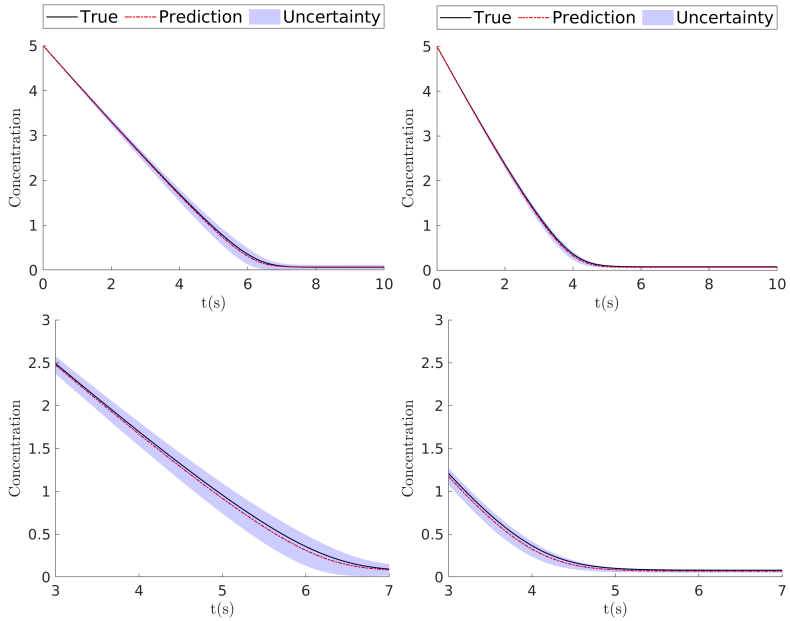

Figure 4. Identified systems of two different set of parameters with 2% noise: (1st column) $[j_x, V_{max}, K_m] = [0.3, 1.2, 0.2]$ and (2nd column) $[j_x, V_{max}, K_m] = [0.3, 1.8, 0.4]$. The trajectory of the state variable x is shown and the propagated uncertainty is given by the 3-σ region. The 2nd row is the zoomed-in view.

3.4. **ICP model discovery.** Lastly, we consider a more challenging case, identifying an idealized intracranial pressure (ICP) dynamics model from multiple synthetic patient-specific datasets. ICP is the pressure inside the skull, and ICP monitoring is essential to manage many cerebral diseases, such as brain injury, hemorrhage, and hydrocephalus. The elevated ICP will lower the perfusion pressure and thus decrease the total cerebral blood flow, which can damage the brain tissue even cause death. The current clinical practice of ICP monitoring is highly-invasive and can cause infection or brain tissue damage. It is preferable to estimate the ICP based on other non-invasively measurable signals related to ICPs. It is known that the ICP dynamics are driven by the interactions of blood flows, cerebrospinal flows, and brain tissues. Clinically, arterial blood pressure (ABP) and cerebral blood flow velocity (CBF) can be monitored in a non-invasive way. If a model that describes the interacting dynamics of intracranial systems is available, we can monitor the ABP and CBF signals to estimate the ICP non-invasively. However, due to the high complexity of intracranial systems, there is no closed-form first-principle model that faithfully describes the ICP dynamics. There is a hope to identify explicit model forms from massive ICP, CBF, and ABP data of a cohort of patients. Under this motivation, we will discover a classical ICP model form, Ursino-Lodi model [36], which simplifies the ICP system as a resistor-capacitor circuit. Note that in current research, we try to identify the classical model based on synthetic data instead of directly identifying dynamical models from real-world data. We acknowledge that directly identifying complex dynamical systems from real-world data is a more important and also more challenging in system identification. In terms of the Ursino-Lodi model, it has two state variables, capacity C_a and ICP (P_{ic}), and two forcing variables, ABP (P_a) and CBF (q). The governing equation of ICP dynamics is given as,

$$\begin{aligned}
\frac{dP_{ic}}{dt} &= \frac{k_E P_{ic}}{1 + C_a k_E P_{ic}} \big[C_a \frac{dP_a}{dt} + \frac{dC_a}{dt}(P_a - P_{ic}) + \frac{P_c - P_{ic}}{R_f} - \frac{P_{ic} - P_{vs}}{R_o} \big], \\
\frac{dC_a}{dt} &= \frac{1}{\tau}[-C_a + \sigma(G \cdot x_a)] \\
x_a &= \frac{q - q_n}{q_n}.
\end{aligned} \tag{24}$$

where k_E is the intracranial elastance coefficient, R_f is cerebrospinal fluid resistance and R_o is outflow resistance. τ is the constant of regulation, σ is a sigmoid static function and G is the maximum autoregulation gain. The q_n is the CBF constant of tissue metabolism and x_a is the normalized CBF changes. To simplify the model forms, we combined the following physiological parameters,

$$\begin{aligned}
P_c &= \frac{P_a R_{pv} + P_{ic} R_a}{R_a + R_{pv}}, \\
R_a &= \frac{k_R C_{an}^2}{V_a^2}, \\
V_a &= C_a (P_a - P_{ic}), \\
q &= \frac{P_a - P_c}{R_a}.
\end{aligned} \tag{25}$$

where R_{pv} and R_a are proximal venous resistance and regulated resistance respectively, k_R is resistance coefficient, C_{an} is the basal arterial compliance, and P_c is the capillary pressure. Then the compliance C_a and resistance R_a can be expressed

as functions of state/forcing variables P_a, P_{ic} and q,

$$\begin{aligned} C_a &= \left[\frac{k_R C_{an}^2 q}{(P_a - P_{ic})^2[(P_a - P_{ic}) - R_{pv}q]}\right]^{0.5}, \\ R_a &= \frac{k_R C_{an}^2}{C_a^2 (P_a - P_{ic})^2}. \end{aligned} \tag{26}$$

Finally, the governing equations of ICP dynamics can be rearranged as,

$$\begin{aligned} \frac{dC_a}{dt} = \frac{k_R C_{an}^2}{2C_a} \frac{1}{[(P_a - P_{ic})^2 (P_a - P_{ic} - R_{pv}q)]^2} \Big(& \frac{dq}{dt}(P_a - P_{ic})^2 (P_a - P_{ic} - R_{pv}q) \\ & - q[(2(P_a - P_{ic})(\frac{dP_a}{dt} - \frac{dP_{ic}}{dt}))(P_a - P_{ic} - R_{pv}q) \\ & + (P_a - P_{ic})^2 (\frac{dP_a}{dt} - \frac{dP_{ic}}{dt} - R_{pv}\frac{dq}{dt})]\Big). \end{aligned} \tag{27}$$

As a result, the ICP system is simplified as a first-order ODE system. The variables ABP (P_a) and CBF (q) are taken as forcing inputs to the system, which can be measured in a non-intrusive manner. Their time derivatives can be calculated either from the finite difference (for clean data) or the denoising neural networks (for noisy measurements). Substituting Eq. 27 into Eq. 24 leads to the final closed equation for ICP dynamics, which is a very complicated implicit ODE system. To deal with the implicit system, the iterative process from the SINDy-PI has to be adopted here, where the left-hand side library is given by Eq 28, with the correct form marked in bold font. The right-hand side library terms, the correct coefficient, and identified terms and coefficients can be found in Table 8.

$$\left[\begin{array}{ccccc} | & | & | & | & | \\ \boldsymbol{R_a \frac{dP_{ic}}{dt}} & \frac{dP_{ic}}{dt} & C_a R_a P_{ic} & C_a P_{ic} \frac{dP_{ic}}{dt} & C_a \frac{dP_{ic}}{dt} \\ | & | & | & | & | \end{array}\right] \tag{28}$$

Synthetic ICP databases are generated by numerically solving the governing ODE systems as described in the original paper [36] with different parameter settings. To be specific, the ODE system uses the 4th order Runge-Kutta method to marching in time, and the simulated time span is $150s$. The initial conditions are given as $P_{ic} = 9.5$ and $C_a = 0.15$. The simulation results are then corrupted by adding 1% noise to the simulated signals P_a, P_{ic}, q and 0.2% noise to their derivatives, serving as the process noise. As described above, the Ursino-Lodi model is a complicated algebraic-differential system and has many control parameters. Here we chose to parametrize three different parameters and generate five different databases, as shown in Table. 7. The three variable parameters are basal arterial compliance (C_{an}), the intracranial elastance coefficient (k_E), and cerebrospinal fluid resistance (R_f). In

Parameter	C_{an}	R_f	k_E
1	0.15	2.38×10^3	0.231
2	0.125	2.38×10^3	0.231
3	0.15	2.5×10^3	0.231
4	0.15	2.38×10^3	0.2
5	0.15	2.38×10^3	0.175

Table 7. Control variables for parametric system identification

this problem, the coefficient magnitude of different candidate function terms could vary by several orders, posing significant challenges on sequential threshold pruning since small but important coefficients can be mistakenly pruned out. As a remedy, we rewrote the discovery problem as,

$$\hat{\dot{\boldsymbol{X}}} = \begin{bmatrix} | & | & | & | & | \\ \boldsymbol{\Pi_1} & \boldsymbol{\Pi_2} & \boldsymbol{\Pi_3} & ... & \boldsymbol{\Pi_m} \\ | & | & | & | & | \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ ... \\ \omega_m \end{bmatrix} \tag{29}$$

where $\boldsymbol{\Pi}_* = \boldsymbol{X}_*/\sigma(\boldsymbol{X}_*)$ and $\boldsymbol{\omega}_* = \boldsymbol{W}_* * \sigma(\boldsymbol{X}_*)$. The transformed equation weights $\boldsymbol{\omega}_*$ will be scaled to the proper range. Finally, we can easily get the true mean values using the reverse transform $\mathbb{E}(\boldsymbol{W}_*) = \mathbb{E}(\boldsymbol{\omega}_*)/\sigma(\boldsymbol{X}_*)$ and $\mathbb{V}(\boldsymbol{W}_*) = \mathbb{V}(\boldsymbol{\omega}_*)/\sigma(\boldsymbol{X}_*)^2$, and then $\sigma(\boldsymbol{W}_*) = \sigma(\boldsymbol{\omega}_*)/\sigma(\boldsymbol{X}_*)$

The true model forms and the discovered model forms are listed in Table 8 and Table 10, respectively. It can be seen that the analytical expression of the Ursino-Lodi model can be accurately discovered, and the mean values of the identified coefficients are also close to the ground truth for all five different parameter sets. The summarized results for metrics comparison (rmse, M_P and M_R) are shown in Table. 9. By forward propagating the predicted uncertainties in the identified coefficients, the propagated uncertainties in the ICP dynamics can be estimated, as shown in Fig. 5. And we only show the UQ result for the group sparsity approach. Consistent with the results in previous cases, the predicted ICP mean coincides well with the true value, and the uncertainty range covers the prediction well. In this case, the stiffness of the ICP dynamics is large, which means a small perturbation of the coefficient can generate a significant variance in the predicted ICP dynamics. Some combination of coefficients can even lead to diverged solutions, which has also been reported in [13]. The diverged samples are eliminated here for the uncertainty propagation.

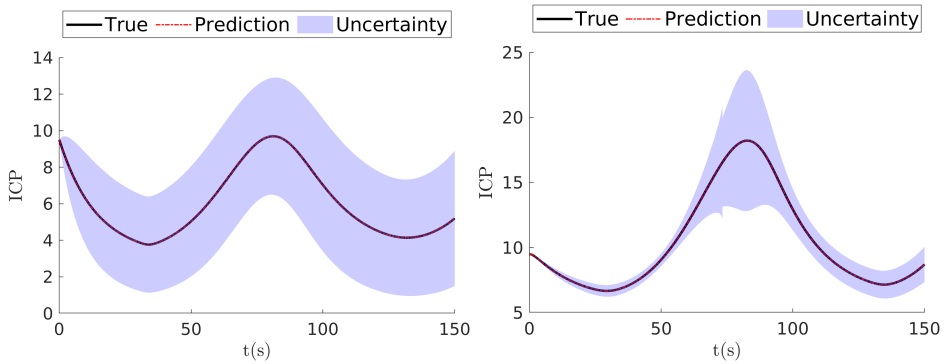

Figure 5. Identified systems of two different set of parameters with 1% noise: (1st column) $[C_{an}, R_f, k_E] = [0.125, 2.38 \times 10^3, 0.231]$ and (2nd column) $[C_{an}, R_f, k_E] = [0.15, 2.38 \times 10^3, 0.175]$. The trajectory of the ICP is shown and the propagated uncertainty is given by the 3-σ region.

True systems						
Term	Coeff	1	2	3	4	5
$P_{ic}P_aR_a\frac{dC_a}{dt}$	λ_1	0.231	0.231	0.231	0.2	0.175
$P_{ic}^2R_a\frac{dC_a}{dt}$	λ_2	-0.231	-0.231	-0.231	-0.2	-0.175
P_aP_{ic}	λ_3	1.2035×10^{-4}	1.2035×10^{-4}	1.1458×10^{-4}	1.0420×10^{-4}	9.118×10^{-5}
P_{ic}^2	λ_4	-1.6571×10^{-4}	-1.6571×10^{-4}	-1.5993×10^{-4}	-1.4347×10^{-4}	-1.2554×10^{-4}
P_{ic}^2Ra	λ_5	-3.6576×10^{-5}	-3.6576×10^{-5}	-3.6576×10^{-5}	-3.1668×10^{-5}	-2.7709×10^{-5}
$P_{ic}Ra$	λ_6	2.1946×10^{-4}	2.1946×10^{-4}	2.1946×10^{-4}	1.9001×10^{-4}	1.6625×10^{-4}
P_{ic}	λ_7	2.7213×10^{-4}	2.7213×10^{-4}	2.7213×10^{-4}	2.3561×10^{-4}	2.0616×10^{-4}
$\frac{dP_{ic}}{dt}$	λ_8	-1.24	-1.24	-1.24	-1.24	-1.24
$C_aP_{ic}R_a\frac{dP_{ic}}{dt}$	λ_9	-0.231	-0.231	-0.231	-0.2	-0.175
$C_aP_{ic}\frac{dP_{ic}}{dt}$	λ_{10}	-0.2864	-0.2864	-0.2864	-0.248	-0.217
$P_aP_{ic}\frac{dC_a}{dt}$	λ_{11}	-0.2864	-0.2864	-0.2864	-0.248	-0.217
$P_{ic}^2\frac{dC_a}{dt}$	λ_{12}	-0.2864	-0.2864	-0.2864	-0.248	-0.217

Table 8. Right hand-side library and true parametric coefficient

Identified systems with group sparsity					
Case	1	2	3	4	5
rmse	406.25	407.79	407.79	359.38	351.82
$\mathbf{M_R}$	1	1	1	1	1
$\mathbf{M_P}$	1	1	1	1	0.92
Identified systems with non-parsimonious model					
rmse	957.18	956.50	957.26	966.70	973.60
$\mathbf{M_R}$	0.79	0.77	0.79	0.77	0.77
$\mathbf{M_P}$	0.92	0.83	0.92	0.83	0.83

Table 9. Different metrics for parametric ICP model

4. **Conclusion.** This work presents a sequential threshold sparse Bayesian learning approach with group sparsity to identify the parsimonious equation forms for a parametric dynamic system using multiple datasets simultaneously. A vital feature of the proposed approach is the group-sparsifying Bayesian learning from multiple datasets governed by the same model form but with different parameters, which is beneficial to identify the most parsimonious model forms. Moreover, to better deal with noisy measurement data and improve derivative computation, a DNN-based denoising method is used for data prepossessing. Several linear/nonlinear ODE systems in both explicit and implicit settings are studied to demonstrate the effectiveness of the proposed method. The numerical results show that the recovered mean is close to the true values. Moreover, the posterior uncertainty for the identified coefficients can be reasonably estimated. The parameter uncertainty can be propagated via the identified dynamical model using the Monte-Carlo method, which can provide uncertainty estimation of the state predictions. Although we have demonstrated the merit of synchronous learning with multiple parametric datasets, the current approach has several limitations, which can be improved in the future. For example, the current method is still less robust for discovering implicit systems when it comes to large data noise. To make further improvement, one potential direction is to couple physics-informed neural networks (PINN) with sparse Bayesian learning techniques, as some recent study of combining PINN and SINDy shows the great capability to deal with high-level data scarcity and large data noise [31, 6].

Identified systems with group sparsity						
Term	Coeff	1	2	3	4	5
$P_{ic}P_aR_a\frac{dC_a}{dt}$	λ_1	0.231	0.231	0.231	0.2	0.175
	$\sigma(\lambda_1)$	2.3745×10^{-4}	5.9450×10^{-4}	2.7723×10^{-4}	3.4780×10^{-4}	3.9709×10^{-4}
$P_{ic}^2R_a\frac{dC_a}{dt}$	λ_2	-0.231	-0.2307	-0.231	-0.2	-0.1748
	$\sigma(\lambda_2)$	5.9808×10^{-4}	2.02×10^{-2}	8.626×10^{-4}	3×10^{-3}	7.6×10^{-3}
P_aP_{ic}	λ_3	1.1972×10^{-4}	1.1905×10^{-4}	1.1421×10^{-4}	1.0319×10^{-4}	8.886×10^{-5}
	$\sigma(\lambda_3)$	2.62×10^{-5}	5.881×10^{-5}	3.841×10^{-5}	6.221×10^{-5}	1.6×10^{-5}
P_{ic}^2	λ_4	-1.6358×10^{-4}	-1.6118×10^{-4}	-1.5867×10^{-4}	-1.4143×10^{-4}	-1.1745×10^{-4}
	$\sigma(\lambda_4)$	6.862×10^{-5}	1.4322×10^{-4}	1.0414×10^{-4}	1.3692×10^{-4}	2.907×10^{-5}
P_{ic}^2Ra	λ_5	-3.6628×10^{-5}	-3.6622×10^{-5}	-3.6604×10^{-4}	-3.1036×10^{-5}	-2.7871×10^{-5}
	$\sigma(\lambda_5)$	2.364×10^{-6}	1.1038×10^{-5}	2.988×10^{-6}	4.857×10^{-5}	6.661×10^{-6}
$P_{ic}Ra$	λ_6	2.2857×10^{-4}	2.3723×10^{-4}	2.2446×10^{-4}	2.0615×10^{-4}	1.9794×10^{-4}
	$\sigma(\lambda_6)$	4.2186×10^{-4}	8.3757×10^{-4}	5.9635×10^{-4}	1.1×10^{-3}	1.8554×10^{-4}
P_{ic}	λ_7	2.1870×10^{-4}	1.5004×10^{-4}	2.4190×10^{-4}	1.3637×10^{-4}	0
	$\sigma(\lambda_7)$	2×10^{-3}	4.7×10^{-3}	3.1×10^{-3}	5.6×10^{-3}	0
$\frac{dP_{ic}}{dt}$	λ_8	-1.2071	-1.1633	-1.2195	-1.1759	-1.0668
	$\sigma(\lambda_8)$	0.9734	1.8386	1.5444	2.4216	0.4493
$C_aP_{ic}R_a\frac{dP_{ic}}{dt}$	λ_9	-0.231	-0.2312	-0.231	-0.2	-0.1751
	$\sigma(\lambda_9)$	5.1×10^{-4}	1.19×10^{-2}	7.4645×10^{-4}	2.4×10^{-3}	5.4×10^{-3}
$C_aP_{ic}\frac{dP_{ic}}{dt}$	λ_{10}	-0.2789	-0.2678	-0.2817	-0.2351	-0.1859
	$\sigma(\lambda_{10})$	0.2246	0.4533	0.3567	0.4868	8.89×10^{-2}
$P_aP_{ic}\frac{dC_a}{dt}$	λ_{11}	-0.2789	-0.2687	-0.2817	-0.2352	-0.1867
	$\sigma(\lambda_{11})$	0.2246	0.4246	0.3562	0.4843	7.77×10^{-2}
$P_{ic}^2\frac{dC_a}{dt}$	λ_{12}	-0.2788	-0.2703	-0.2817	-0.2352	-0.1877
	$\sigma(\lambda_{12})$	0.2249	0.3936	0.3562	0.4819	7.36×10^{-2}

Table 10. Right hand-side library and inferred parametric coefficient, where λ_i denotes the coefficient and $\sigma(\lambda_i)$ denotes standard deviation for the parsimonious library terms.

Acknowledgment. The authors would like to acknowledge the funds from National Science Foundation under award numbers CMMI-1934300 and OAC-2047127, and startup funds from the College of Engineering at University of Notre Dame in supporting this study.

Compliance with Ethical Standards. Conflict of Interest: The authors declare that they have no conflict of interest.

REFERENCES

[1] C. M. Bishop and M. Tipping, Variational relevance vector machines, arXiv:1301.3838.
[2] S. L. Brunton, J. L. Proctor and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proceedings of the National Academy of Sciences*, **113** (2016), 3932-3937.
[3] K. P. Champion, S. L. Brunton and J. N. Kutz, Discovery of nonlinear multiscale systems: Sampling strategies and embeddings, *SIAM Journal on Applied Dynamical Systems*, **18** (2019), 312-333.
[4] K. Champion, B. Lusch, J. N. Kutz and S. L. Brunton, Data-driven discovery of coordinates and governing equations, *Proceedings of the National Academy of Sciences*, **116** (2019), 22445-22451.
[5] Z. Chen, Y. Liu and H. Sun, Physics-informed learning of governing equations from scarce data, *Nature Communications*, **12** (2021), 1-13.
[6] Z. Chen, Y. Liu and H. Sun, Physics-informed learning of governing equations from scarce data, arXiv:2005.03448.
[7] H. K. Chu and M. Hayashibe, Discovering interpretable dynamics by sparsity promotion on energy and the lagrangian, *IEEE Robotics and Automation Letters*, **5** (2020), 2154-2160.
[8] M. Corbetta, Application of sparse identification of nonlinear dynamics for physics-informed learning, in *2020 IEEE Aerospace Conference*, IEEE, (2020), 1-8.

[9] A. C. Faul and M. E. Tipping, A variational approach to robust regression, in *International Conference on Artificial Neural Networks*, Springer, (2001), 95-102.

[10] A. C. Faul and M. E. Tipping, Analysis of sparse Bayesian learning, in *Advances in Neural Information Processing Systems*, (2002), 383-389.

[11] H. Gao, L. Sun and J.-X. Wang, PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain, *Journal of Computational Physics*, **428** (2021), 110079.

[12] X. Han, H. Gao, T. Pfaff, J.-X. Wang and L. Liu, Predicting physics in mesh-reduced space with temporal attention, in *International Conference on Learning Representations*, 2022.

[13] S. M. Hirsh, D. A. Barajas-Solano and J. N. Kutz, Sparsifying priors for bayesian uncertainty quantification in model discovery, `arXiv:2107.02107`.

[14] K. Kaheman, J. N. Kutz and S. L. Brunton, SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics, *Proceedings of the Royal Society A*, **476** (2020), 20200279.

[15] E. Kaiser, J. N. Kutz and S. L. Brunton, Sparse identification of nonlinear dynamics for model predictive control in the low-data limit, *Proceedings of the Royal Society A*, **474** (2018), 20180335.

[16] S. Kim, P. Y. Lu, S. Mukherjee, M. Gilbert, L. Jing, V. Čeperić and M. Soljačić, Integration of neural network-based symbolic regression in deep learning for scientific discovery, *IEEE Transactions on Neural Networks and Learning Systems*, **32** (2020), 4166-4177.

[17] X.-Y. Liu and J.-X. Wang, Physics-informed dyna-style model-based deep reinforcement learning for dynamic control, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **477** (2021), 20210618.

[18] Z. Long, Y. Lu and B. Dong, Pde-net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network, *Journal of Computational Physics*, **399** (2019), 108925.

[19] L. Lu, P. Jin, G. Pang, Z. Zhang and G. E. Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators, *Nature Machine Intelligence*, **3** (2021), 218-229.

[20] N. M. Mangan, T. Askham, S. L. Brunton, J. N. Kutz and J. L. Proctor, Model selection for hybrid dynamical systems via sparse regression, *Proceedings of the Royal Society A*, **475** (2019), 20180534.

[21] N. M. Mangan, S. L. Brunton, J. L. Proctor and J. N. Kutz, Inferring biological networks by sparse identification of nonlinear dynamics, *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, **2** (2016), 52-63.

[22] B. K. Nelson, Time series analysis using autoregressive integrated moving average (ARIMA) models, *Academic Emergency Medicine*, **5** (1998), 739-744.

[23] W. Pan, A. Sootla and G.-B. Stan, Distributed reconstruction of nonlinear networks: An ADMM approach, *IFAC Proceedings Volumes*, **47** (2014), 3208-3213.

[24] W. Pan, Y. Yuan, J. Gonçalves and G.-B. Stan, A sparse Bayesian approach to the identification of nonlinear state-space systems, *IEEE Transactions on Automatic Control*, **61** (2015), 182-187.

[25] W. Pan, Y. Yuan, L. Ljung, J. Gonçalves and G.-B. Stan, Identification of nonlinear state-space systems from heterogeneous datasets, *IEEE Transactions on Control of Network Systems*, **5** (2017), 737-747.

[26] L. Piroddi, M. Farina and M. Lovera, Polynomial narx model identification: a Wiener–Hammerstein benchmark, *IFAC Proceedings*, **42** (2009), 1074-1079.

[27] M. Quade, M. Abel, J. Nathan Kutz and S. L. Brunton, Sparse identification of nonlinear dynamics for rapid model recovery, Chaos: An Interdisciplinary, *Journal of Nonlinear Science*, **28** (2018), 063116.

[28] S. H. Rudy, S. L. Brunton, J. L. Proctor and J. N. Kutz, Data-driven discovery of partial differential equations, *Science Advances*, **3** (2017), e1602614.

[29] S. H. Rudy, J. N. Kutz and S. L. Brunton, Deep learning of dynamics and signal-noise decomposition with time-stepping constraints, *Journal of Computational Physics*, **396** (2019), 483-506.

[30] S. Rudy, A. Alla, S. L. Brunton and J. N. Kutz, Data-driven identification of parametric partial differential equations, *SIAM Journal on Applied Dynamical Systems*, **18** (2019), 643-660.

[31] F. Sun, Y. Liu and H. Sun, Physics-informed spline learning for nonlinear dynamics discovery, `arXiv:2105.02368`.

[32] L. Sun, H. Gao, S. Pan and J.-X. Wang, Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data, *Computer Methods in Applied Mechanics and Engineering*, **361** (2020), 112732.
[33] L. Sun and J.-X. Wang, Physics-constrained Bayesian neural network for fluid flow reconstruction with sparse and noisy data, *Theoretical and Applied Mechanics Letters*, **10** (2020), 161-169.
[34] M. E. Tipping, Sparse Bayesian learning and the relevance vector machine, *Journal of Machine Learning Research*, **1** (2001), 211-244.
[35] M. E. Tipping, A. C. Faul, et al., Fast marginal likelihood maximisation for sparse Bayesian models, in *AISTATS*, 2003.
[36] M. Ursino and C. A. Lodi, A simple mathematical model of the interaction between intracranial pressure and cerebral hemodynamics, *Journal of Applied Physiology*, **82** (1997), 1256-1269.
[37] S. Wang, H. Wang and P. Perdikaris, Learning the solution operator of parametric partial differential equations with physics-informed deeponets, *Science Advances*, **7** (2021), eabi8605.
[38] J. Wang, X. Xie, J. Shi, W. He, Q. Chen, L. Chen, W. Gu and T. Zhou, Denoising autoencoder, a deep learning algorithm, aids the identification of a novel molecular signature of lung adenocarcinoma, *Genomics, Proteomics & Bioinformatics*, **18** (2020), 468-480.
[39] H. Wu, P. Du, R. Kokate and J.-X. Wang, A semi-analytical solution and AI-based reconstruction algorithms for magnetic particle tracking, *Plos One*, **16** (2021), e0254051.
[40] Y. Yang, M. Aziz Bhouri and P. Perdikaris, Bayesian differential programming for robust systems identification under uncertainty, *Proceedings of the Royal Society A*, **476** (2020), 20200290.
[41] L. Zhang and H. Schaeffer, On the convergence of the SINDy algorithm, *Multiscale Modeling & Simulation*, **17** (2019), 948-972.
[42] R. Zhang, Z. Chen, S. Chen, J. Zheng, O. Büyüköztürk and H. Sun, Deep long short-term memory networks for nonlinear structural seismic response prediction, *Computers & Structures*, **220** (2019), 55-68.
[43] S. Zhang and G. Lin, Robust data-driven discovery of governing physical laws with error bars, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **474** (2018), 20180305.
[44] S. Zhang and G. Lin, Robust data-driven discovery of governing physical laws using a new subsampling-based sparse bayesian method to tackle four challenges (large noise, outliers, data integration, and extrapolation), `arXiv:1907.07788`.
[45] Z. Zhang and Y. Liu, Parsimony-enhanced sparse bayesian learning for robust discovery of partial differential equations, *Mechanical Systems and Signal Processing*, **171** (2022), 108833.

Received February 2022; 1^{st} revision October 2022; Final revision November 2022; Early access December 2022.

Appendix.

rmse	Linear	Cubic	MichaelisMenten	ICP
Parsimonious	5s	13s	4882s	1148s
Non-parsimonious	0.67s	0.2s	1251s	1134s

Table 11. Training cost comparison

Identified systems with group sparsity									
Case	1	2	3	4	5	6	7	8	9
λ_1	-0.0506	-0.0513	-0.0522	-0.1026	-0.1042	-0.1033	-0.1517	-0.1502	-0.1539
$\sigma(\lambda_1)$	0.0038	0.0038	0.0038	0.0033	0.0033	0.0033	0.0030	0.0030	0.0030
λ_2	1.5001	1.9977	2.4989	1.5034	2.0012	2.4992	1.5012	2.0017	2.4964
$\sigma(\lambda_2)$	0.0038	0.0037	0.0037	0.0033	0.0032	0.0032	0.0029	0.0029	0.0029
λ_3	-1.4976	-1.9982	-2.4940	-1.4963	-1.9945	-2.4957	-1.4996	-1.9973	-2.4997
$\sigma(\lambda_3)$	0.0038	0.0038	0.0038	0.0033	0.0033	0.0033	0.0030	0.0030	0.0030
λ_4	-0.0510	-0.0494	-0.0474	-0.1013	-0.0992	-0.0989	-0.1536	-0.1535	-0.1492
$\sigma(\lambda_4)$	0.0038	0.0037	0.0037	0.0033	0.0032	0.0032	0.0029	0.0029	0.0029
Identified systems without group sparsity									
λ_1	-0.0487	-0.0487	-0.0489	-0.1006	-0.1014	-0.1000	-0.1498	-0.1476	-0.1509
$\sigma(\lambda_1)$	0.0038	0.0038	0.0038	0.0033	0.0033	0.0033	0.0030	0.0030	0.0030
λ_2	1.5005	1.9995	2.5017	1.5017	2.0008	2.5000	1.4982	1.9999	2.4958
$\sigma(\lambda_2)$	0.0038	0.0037	0.0037	0.0033	0.0032	0.0032	0.0029	0.0029	0.0029
$C(\text{const1})$	-0.0626	-0.0847	-0.1048	-0.0436	-0.0608	-0.0740	0	-0.0475	-0.0597
$\sigma(C(\text{const1}))$	0.0035	0.0035	0.0035	0.0026	0.0026	0.0026	0	0.0021	0.0021
λ_3	-1.4997	-2.0004	-2.4986	-1.4992	-1.9987	-2.4994	-1.5012	-1.9999	-2.5016
$\sigma(\lambda_3)$	0.0019	0.0019	0.0019	0.0017	0.0016	0.0016	0.0015	0.0015	0.0015
λ_4	-0.0507	-0.0506	-0.0502	-0.0996	-0.0993	-0.0999	-0.1499	-0.1507	-0.1493
$\sigma(\lambda_4)$	0.0019	0.0019	0.0019	0.0016	0.0016	0.0016	0.0015	0.0015	0.0015
$C(\text{const2})$	0	0.0440	0.0551	0	0	0.0411	0	0	0
$\sigma(C(\text{const2}))$	0	0.0017	0.0017	0	0	0.0013	0	0	0
True systems									
λ_1	-0.05	-0.05	-0.05	-0.1	-0.1	-0.1	-0.15	-0.15	-0.15
λ_2	1.5	2.0	2.5	1.5	2.0	2.5	1.5	2.0	2.5
λ_3	-1.5	-2.0	-2.5	-1.5	-2.0	-2.5	-1.5	-2.0	-2.5
λ_4	-0.05	-0.05	-0.05	-0.1	-0.1	-0.1	-0.15	-0.15	-0.15

Table 12. System identification results for the linear parametric ODE, $\frac{dx}{dt} = \lambda_1 x + \lambda_2 y$, state $\frac{dy}{dt} = \lambda_3 x + \lambda_4 y$ with 10% data noise, where λ_i denotes the relevant coefficient, $C(\text{const1})$ denotes the redundant *constant* coefficient and $\sigma(C(\text{const1}))$ denotes the standard deviation for equation $\frac{dx}{dt} = \lambda_1 x + \lambda_2 y$, while $C(\text{const2})$ and $\sigma(C(\text{const2}))$ are corresponding values for equation $\frac{dy}{dt} = \lambda_3 x + \lambda_4 y$.

Identified systems with group sparsity									
Case	1	2	3	4	5	6	7	8	9
y^2	-0.1749	-0.2424	-0.2813	-0.1295	-0.1940	-0.2175	-0.1045	-0.1744	-0.2075
$\sigma(y^2)$	0.0044	0.0061	0.0088	0.0027	0.0041	0.0050	0.0026	0.0035	0.0051
λ_1	-0.0485	-0.0458	-0.0503	-0.1007	-0.0978	-0.1025	-0.1498	-0.1536	-0.1525
$\sigma(\lambda_1)$	0.0039	0.0053	0.0075	0.0025	0.0038	0.0045	0.0025	0.0033	0.0047
λ_2	1.5013	1.9995	2.4978	1.4994	2.0097	2.4998	1.4909	2.0109	2.5204
$\sigma(\lambda_2)$	0.0037	0.0052	0.0075	0.0023	0.0036	0.0044	0.0022	0.0031	0.0045
x^2	0.1797	0.2177	0.3016	0.1434	0.1922	0.2480	0.1273	0.1710	0.2083
$\sigma(x^2)$	0.0047	0.0060	0.0084	0.0032	0.0049	0.0061	0.0030	0.0037	0.0060
λ_3	-1.5033	-2.0000	-2.5049	-1.5059	-1.9995	-2.5155	-1.5117	-2.0026	-2.5049
$\sigma(\lambda_3)$	0.0041	0.0051	0.0072	0.0029	0.0043	0.0054	0.0028	0.0034	0.0055
λ_4	-0.0508	-0.0541	-0.0514	-0.0998	-0.1056	-0.0950	-0.1473	-0.1528	-0.1485
$\sigma(\lambda_4)$	0.0039	0.0050	0.0072	0.0026	0.0041	0.0053	0.0023	0.0030	0.0053
Identified systems without group sparsity									
$C(y^2)$	-0.1772	-0.2423	-0.2791	-0.1377	-0.2122	-0.2376	-0.1153	-0.1898	-0.1975
$\sigma(C(y^2))$	0.0056	0.0055	0.0082	0.0034	0.0042	0.0056	0.0030	0.0039	0.0062
λ_1	-0.0489	-0.0468	-0.0514	-0.0990	-0.0931	-0.1005	-0.1476	-0.1493	-0.1548
$\sigma(\lambda_1)$	0.0037	0.0047	0.0055	0.0024	0.0030	0.0038	0.0022	0.0030	0.0045
$C(x^2y)$	0	0	0	0	0	0	0	0	-0.0558
$\sigma(C(x^2y))$	0	0	0	0	0	0	0	0	0.0189
λ_2	1.5068	1.9985	2.5174	1.4950	2.0084	2.5035	1.4918	2.0150	2.5325
$\sigma(\lambda_2)$	0.0051	0.0069	0.0083	0.0034	0.0041	0.0050	0.0022	0.0039	0.0062
$C(x^2y^2)$	0	0	0	0	0.0361	0.0378	0	0.0365	0
$\sigma(C(x^2y^2))$	0	0	0	0	0.0049	0.0062	0	0.0051	0
$C(x^4y)$	0	0	0.0513	0	0	0	0	0	0
$\sigma(C(x^4y))$	0	0	0.0137	0	0	0	0	0	0
$C(x^2y^3)$	0	0	-0.0589	0	0	0	0	0	0
$\sigma(C(x^2y^3))$	0	0	0.0125	0	0	0	0	0	0
$C(x^2)$	0.1798	0.2060	0.2977	0.1418	0.1946	0.2289	0.1264	0.1610	0.2011
$\sigma(C(x^2))$	0.0050	0.0072	0.0092	0.0039	0.0042	0.0058	0.0023	0.0032	0.0059
λ_3	-1.4942	-1.9919	-2.4866	-1.5093	-2.0194	-2.5208	-1.5049	-2.0173	-2.5176
$\sigma(\lambda_3)$	0.0050	0.0063	0.0084	0.0037	0.0039	0.0052	0.0022	0.0031	0.0056
$C(xy^2)$	0	0	0	0	0	0	0	-0.0386	0
$\sigma(C(xy^2))$	0	0	0	0	0	0	0	0.0057	0
λ_4	-0.0511	-0.0548	-0.0502	-0.0994	-0.1089	-0.0977	-0.1472	-0.1549	-0.1487
$\sigma(\lambda_4)$	0.0035	0.0045	0.0062	0.0025	0.0027	0.0039	0.0016	0.0020	0.0042
$C(x^3y)$	0	0	0	0	0	0.0482	0	0	-0.0603
$\sigma(C(x^3y))$	0	0	0	0	0	0.0064	0	0	0.0068
$C(x^2y^2)$	0	0	0	0	0	0	0	0.0368	0
$\sigma(C(x^2y^2))$	0	0	0	0	0	0	0	0.0044	0
$C(xy^3)$	0	0	0	0	0	0	0	0	0.0420
$\sigma(C(xy^3))$	0	0	0	0	0	0	0	0	0.0071
$C(x^3y^2)$	0	0	0	0	0.0501	0	0	0.0637	0
$\sigma(C(x^3y^2))$	0	0	0	0	0.0073	0	0	0.0065	0
True systems									
λ_1	-0.05	-0.05	-0.05	-0.1	-0.1	-0.1	-0.15	-0.15	-0.15
λ_2	1.5	2.0	2.5	1.5	2.0	2.5	1.5	2.0	2.5
λ_3	-1.5	-2.0	-2.5	-1.5	-2.0	-2.5	-1.5	-2.0	-2.5
λ_4	-0.05	-0.05	-0.05	-0.1	-0.1	-0.1	-0.15	-0.15	-0.15

Table 13. System identification results for the parametric cubic systems $\frac{dx}{dt} = \lambda_1 x^3 + \lambda_2 y^3$, state $\frac{dy}{dt} = \lambda_3 x^3 + \lambda_4 y^3$ with 10% data noise.