A Bayesian Approach for Spatio-Temporal Data-Driven Dynamic Equation Discovery[*]

Joshua S. North[†], Christopher K. Wikle[‡], and Erin M. Schliep[§]

Abstract. Differential equations based on physical principals are used to represent complex dynamic systems in all fields of science and engineering. When known, these equations have been shown to well represent real-world dynamics. However, since the true dynamics of complex systems are generally unknown, learning the governing equations can improve our understanding of the mechanisms driving the systems. Here, we develop a Bayesian approach to data-driven discovery of nonlinear spatio-temporal dynamic equations. Our approach can accommodate measurement error and missing data, both of which are common in real-world data, and accounts for parameter uncertainty. The proposed framework is illustrated using three simulated systems with varying amounts of measurement uncertainty and missing data and applied to a real-world system to infer the temporal evolution of the vorticity of the streamfunction.

Keywords: Bayesian dynamic discovery, data-driven discovery, nonlinear dynamic equation, partial differential equation.

1 Introduction

Dynamic equations parameterized by partial differential equations (PDEs) – equations relating a partial derivative of a variable to a function of its current state – are used across all fields of science and engineering to describe complex processes. Dynamic equations encode physical processes by a set of mathematical equations, enabling complex systems such as the spread of infectious disease (Bolker and Grenfell, 1995; Mangal et al., 2008; Kühnert et al., 2014), evolution of invasive species (Hastings, 1996; Liu et al., 2019), weather and climate (Charney et al., 1950; Holton and Hakim, 2012), and the flow of fluids (White and Majdalani, 2006) to be characterized and modeled (see also Higham et al., 2016, for further discussion). Given that any real world process is only approximately characterized by mathematical relationships, mathematically derived dynamic equations are inherently unable to completely characterize a real world

†Climate and Ecosystem Sciences Division, Lawrence Berkeley National Laboratory, Berkeley, CA, 1 Cyclotron Road, USA, jsnorth@lbl.gov
‡Department of Statistics, University of Missouri, Columbia, MO, 146 Middlebush Hall, USA, wiklec@missouri.edu
§Department of Statistics, North Carolina State University, Raleigh, NC, 2311 Stinson Drive, USA, emschliep@ncsu.edu

system. This suggests that observations of the real world system could be used to better inform the underlying dynamic equations.

Recently, there has been a push to use data to discover the governing equations in complex systems. Originally proposed using symbolic regression (Bongard and Lipson, 2007; Schmidt and Lipson, 2009), the focus has since shifted to sparse regression and deep modeling. The original sparse regression approach, termed *Sparse Identification of Nonlinear Dynamics* (SINDy; Brunton et al., 2016), used numerical differentiation to construct a response that is regressed against a library of functions that potentially govern the system – a library is simply a list of functions (library terms), such as derivatives in spatial or temporal dimensions, that potentially relate the current state of the system to its partial derivative. Through sparse regression, either with an ℓ_1 penalization term (Tibshirani, 1996) or using a thresholding approach (Zheng et al., 2019; Champion et al., 2020), key terms governing a variety of ordinary differential equations (ODEs) are identified. The SINDy framework was extended to include PDEs and parametric forms (Schaeffer, 2017; Rudy et al., 2017, 2019a,b), stochastic dynamical systems (Boninsegna et al., 2018), uncertainty quantification of the parameters (Zhang and Lin, 2018; Yang et al., 2019; Niven et al., 2020; Fasel et al., 2021; Hirsh et al., 2021), and has been incorporated into a Python package (PySINDy; de Silva et al., 2020).

Deep models used for data-driven discovery of dynamics can broadly be grouped into two categories – approximating dynamics (Raissi et al., 2017; Raissi and Karniadakis, 2018; Raissi et al., 2020; Sun et al., 2019; Wu and Xiu, 2020) and discovering dynamics (Both et al., 2021; Xu et al., 2019, 2020, 2021; Long et al., 2017, 2019). Using deep models to approximate the dynamics of complex systems enables a computationally inexpensive method to obtain measurements of otherwise difficult to simulate systems while still obeying physical principles (see Reichstein et al., 2019, for an in-depth discussion on the topic). However, our goal is the discovery of the governing equations where "data-driven discovery" refers to the discovery of the *functional form* of the system; there are various deep modeling approaches that have also been used in this context. Combining deep modeling and sparse identification, Both et al. (2021) approximate the PDE using a neural network, which is used to compute derivatives and construct a sparse formulation similar to the SINDy approach. Long et al. (2017, 2019) use a symbolic neural network, an extension of symbolic regression, and a numerical approximation of differential operators in a feed-forward network to discover PDEs and Xu et al. (2021) use a fully connected neural network with a genetic algorithm to express and generate terms of a PDE.

Two open problems in data-driven discovery are (i) accounting for measurement uncertainty (i.e., missing data and measurement error) and (ii) parameter uncertainty. Existing methods that extend the SINDy framework to account for uncertainty quantification employ either a bootstrap approach (Fasel et al., 2021) or a Bayesian approach with variable shrinkage/selection priors placed on the coefficients associated with the library terms (Zhang and Lin, 2018; Niven et al., 2020; Hirsh et al., 2021). These approaches directly follow the first steps of the SINDy framework, where derivatives are computed numerically, data is de-noised, and the feature library is constructed. In this manner, the removal of the uncertainty associated with the observed data is decoupled

from the modeling process (i.e., not model based); the estimate for the system uncertainty is then dependent on the numerical differentiation method, which subsequently influences the estimate of the parameter uncertainty. Yang et al. (2020) developed a method to jointly account for uncertainty in the observed data and parameters based on differential Bayesian programming. While this approach directly accounts for measurement uncertainty, it requires derivatives be computed using a numerical solver (e.g., Runge-Kutta), which can lead to numerical instabilities and cannot accommodate missing data.

To account for measurement uncertainty and missing data when modeling complex nonlinear systems, statisticians have incorporated dynamic equations parameterized by PDEs into Bayesian hierarchical models (BHM; Berliner, 1996; Royle et al., 1999; Wikle et al., 2001). These models, sometimes called physical statistical models (PSM), enable modeling mechanistic relationships within a probabilistic framework (see Berliner, 2003; Cressie and Wikle, 2011; Kuhnert, 2017, for an overview). PSMs are composed of three sub-models – data, process, and parameter models. To account for measurement and mechanistic uncertainty, PSMs consider the dynamics to be latent in the process stage, and represent the observed data in the data stage conditioned on these latent dynamics. While PSMs have been used to model and better understand complex systems, such as ocean surface winds (Wikle et al., 2001; Milliff et al., 2011) and the spread of avian species (Wikle, 2003; Hooten and Wikle, 2008), they require the dynamic relationships (although not weights/parameters associated with those relationships) to be specified *a priori*. To increase flexibility for representing complex processes, PSMs consider the parameters that describe the influence of dynamic components to be random, and often allow them to have spatial or temporal dependence, enabling the data to inform the model. While PSMs are adaptable to a variety of problems and provide inference on how the process may be evolving, they cannot be used to discover new dynamical relationships.

Recently, North et al. (2022) proposed a Bayesian data-driven discovery method that accounts for measurement and parameter uncertainty using a BHM framework composed of data, process, and parameter models for ODEs. Analogous to PSMs, the dynamics are modeled as a latent process and measurement error is accounted for in the data model. Allowing the dynamics to be a latent random process is different than previous data-driven discovery methods that attempt to quantify uncertainty. To link the dynamic system to its derivatives probabilistically, the dynamic process and all the derivatives are modeled using a basis expansion with a common set of basis functions. Derivatives are then obtained analytically using the basis expansion, such that dependence between the dynamic process and its derivatives is explicit. A library of potential functions can be constructed based on the basis coefficients and functions, and a variable selection prior can identify the key functions governing the nonlinear system.

Here, we propose a spatio-temporal extension to Bayesian data-driven discovery for PDEs. The general framework follows North et al. (2022), however the addition of the spatial dimension requires a complete reformulation of the process model. To account for the extra dimension (i.e., space), we model the dynamic process as a higher-order tensor where the dimensions represent space, time, and the number of components (sometimes

called the system states) in the system. The tensor is decomposed using differentiable basis functions in space and time, probabilistically linking the dynamic system with its spatial and temporal derivatives. The basis decomposition is incorporated into the BHM, enabling potential functions to be constructed using the basis functions and coefficients. A variable selection prior on the coefficients produces a sparse solution set of the resulting system. In contrast to the ODE discovery problem, the library of potential functions for PDEs can exhibit strong multicollinearity. An additional contribution of this work is an approach to account for multicollinearity in high dimensional dynamic systems.

We demonstrate our method using data generated from Burgers' equation, the heat equation, and a predator-prey reaction-diffusion equation with varying levels of measurement error. In addition, we demonstrate our model's ability to accommodate missing data using Burgers' equation. The simulations show that our approach is robust to measurement error and missing data, able to learn the dynamics of complex systems, and provides formal uncertainty quantification on parameter estimates and the confidence of the discovered dynamics. Last, we apply our method to infer the evolution of atmospheric vorticity over time having only observed the streamfunction and obtain results that coincide with geophysical balances (i.e., the barotropic vorticity equation).

The remainder of this paper is organized as follows. In Section 2 we define the tensor and derivative notation used throughout the manuscript. In Section 3 we give background on the general dynamic system, showcase how inference on the derivative of the system is made, and present the Bayesian hierarchical model. In Section 4 we describe parameter estimation and discuss modeling choices. In Section 5 we demonstrate our method on multiple simulated data sets and in Section 6 we perform inference on a real-world system. Section 7 concludes the paper.

2 Preliminary Notation

In this section we define tensor and derivative notation. All variables in this section are used only for illustrating notation. Problem specific notation will be introduced in Section 3.

2.1 Tensor Notation

PDEs are commonly defined over multiple dimensions (e.g., space, time, components), and benefit from the use of higher-order tensor notation when the number of dimensions is three or more. We generally follow the notation of Kolda and Bader (2009) and refer the reader to their work for more details and references of tensor notation and applications.

Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ be a tensor of order N where the $(i_1, i_2, \ldots, i_N)$ element is denoted by $x(i_1, i_2, \ldots, i_N)$. A *slice* of the tensor is a two-dimensional section where all but two indices are held constant. For example, the horizontal, lateral, and frontal slices of the third order tensor $\mathcal{Y} \in \mathbb{R}^{I \times J \times K}$ are denoted by $\mathbf{Y}_{i::}$, $\mathbf{Y}_{:j:}$, and $\mathbf{Y}_{::k}$, respectively. A tensor can be converted to a matrix using n-mode matricization (also

known as unfolding or flattening). The n-mode matricization of the tensor $\mathcal{X}$, denoted by $\mathbf{X}_{(n)}$, arranges the mode-n fibers (the higher-order equivalent of matrix rows and columns) to be columns in the resulting matrix. For example, the possible modes of $\mathcal{Y}$ are $\mathbf{Y}_{(1)} \in \mathbb{R}^{I \times (J \times K)}$, $\mathbf{Y}_{(2)} \in \mathbb{R}^{J \times (I \times K)}$, and $\mathbf{Y}_{(3)} \in \mathbb{R}^{K \times (I \times J)}$. In general, we will only be concerned with the mode-3 matricization of a tensor and will denote $\mathbf{Y}$ in place of $\mathbf{Y}_{(3)}$ (all other modes will be properly denoted).

To multiply a tensor by a matrix $\mathbf{B} \in \mathbb{R}^{J \times I_n}$, we use the n-mode product (i.e., multiply a tensor by a matrix or vector in mode n). The n-mode product of the tensor $\mathcal{X}$ and matrix $\mathbf{B}$ is denoted as $\mathcal{X} \times_n \mathbf{B}$ and is of size $I_1 \times I_2 \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$. Equivalently, in terms of unfolded (matricized) tensors, $\mathcal{Z} = \mathcal{X} \times_n \mathbf{B} \Leftrightarrow \mathbf{Z}_{(n)} = \mathbf{BX}_{(n)}$.

2.2   Tensor Basis Representation

Let $\mathcal{Y} \in \mathbb{R}^{I \times J \times K}$ be the order 3 tensor from Section 2.1. Define the expansion of the (i, j, k) element of $\mathcal{Y}$ as

$$y(i, j, k) \equiv \sum_{p=1}^{I} \sum_{q=1}^{J} \sum_{r=1}^{K} g(p, q, r) a(i, p) b(j, q) c(k, r),$$

where $\{a(i, p) : p = 1, \ldots, I\}$, $\{b(j, q) : q = 1, \ldots, J\}$, and $\{c(k, r) : r = 1, \ldots, K\}$ are basis functions and $\{g(p, q, r) : p = 1, \ldots, I, q = 1, \ldots, J, r = 1, \ldots, K\}$ is the tensor of associated basis coefficients. To reduce the dimension, we keep the first $P < I, Q < J$, and $R < K$ terms from a, b, and c, and define each basis function at values $p = 1, \ldots, P$, $q = 1, \ldots, Q$, and $r = 1, \ldots, R$, respectively. That is, we obtain the approximation

$$\mathcal{Y} \approx \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g(p, q, r) \mathbf{a}(p) \circ \mathbf{b}(q) \circ \mathbf{c}(r) = [\![\mathcal{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C},$$

where $\circ$ is the vector outer product, $\mathcal{G}$ is a $P \times Q \times R$ tensor, $\mathbf{A}$ is a $I \times P$, $\mathbf{B}$ is a $J \times Q$, and $\mathbf{C}$ is a $K \times R$ matrix of basis coefficients where each column is given by $\mathbf{a}(p) \equiv (a(1, p), \ldots, a(I, p))$, $\mathbf{b}(q) \equiv (b(1, q), \ldots, a(J, q))$, $\mathbf{c}(r) \equiv (c(1, r), \ldots, c(K, r))$, and $[\![\mathcal{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ is shorthand notation introduced in Kolda (2006). Our basis decomposition is similar to the Tucker decomposition (Tucker, 1966), except we assume $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ are known and our goal is to estimate $\mathcal{G}$. Note, we provide the expansion only for an order 3 tensor (sufficient for this manuscript), but the concept can be extended to higher order tensors.

2.3   Derivative Notation

As discussed in Section 1, we propose a method to discover the governing equations in PDEs. As the name suggests, a PDE is composed of partial derivatives of some variable $u = u(x, y, t)$ that is indexed in space or time or both. We denote partial derivatives using a subscript, for example $\frac{\partial u}{\partial t} = u_t$, $\frac{\partial u}{\partial x} = u_x$, $\frac{\partial^2 u}{\partial t^2} = u_{tt}$, and so forth. We denote the ith

order of a derivative generally as $\frac{\partial^{(i)}t}{\partial t^{(i)}} = u_{t^{(i)}}$. In order to disambiguate notation, we denote the index of a vector/matrix/tensor using parentheses (e.g., $a(i), a(i,j), a(i,j,k)$), reserving the subscript to denote derivatives.

Within the PDE literature there are different choices of notation to denote the same operation. For example, the Laplacian operator can be denoted as $\Delta u = \nabla^2 u = \nabla\cdot\nabla u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = u_{xx} + u_{yy}$. Wherever an operator such as the Laplacian is used for the first time, we will define it. This may result in our notation differing from other texts, but we aim to be consistent within the paper.

3   Bayesian Dynamic Equation Discovery

Here we propose a general hierarchical model for making inference on nonlinear spatio-temporal dynamic systems. We begin by motivating the general class of PDEs and manipulate them to fit within a statistical framework.

3.1   Dynamic Equations

Consider the general PDE dynamic system describing the evolution of a continuous field $\{\mathbf{u}(\mathbf{s},t) : \mathbf{s} \in D_s, t \in D_t\}$,

$$\mathbf{u}_{t^{(J)}}(\mathbf{s},t) = \mathcal{M}(\mathbf{u}(\mathbf{s},t), \mathbf{u}_x(\mathbf{s},t), \mathbf{u}_y(\mathbf{s},t), \mathbf{u}_{xy}(\mathbf{s},t), \ldots, \mathbf{u}_{t^{(1)}}(\mathbf{s},t), \ldots, \mathbf{u}_{t^{(J-1)}}(\mathbf{s},t), \boldsymbol{\omega}(\mathbf{s},t)), \tag{3.1}$$

where the vector $\mathbf{u}(\mathbf{s},t) \in \mathbb{R}^N$ denotes the realization of the N-dimensional system at location $\mathbf{s}$ and time t (e.g., $\mathbf{u}(\mathbf{s},t) = [u(\mathbf{s},t,1), u(\mathbf{s},t,2), \ldots, u(\mathbf{s},t,N)]')$, $\mathcal{M}(\cdot)$ represents the (potentially nonlinear) evolution function, and $\boldsymbol{\omega}(\mathbf{s},t)$ represents any covariates that might be included in the system. Here, $\mathbf{s} \in \{\mathbf{s}_1, \ldots, \mathbf{s}_S\} = D_s$ is a spatial location in the domain with $|D_s| = S$, and $t \in \{1, \ldots, T\} = D_t$ is the temporal realization of the system where $|D_t| = T$. We define (3.1) in two dimensions (i.e., $D_s \in \mathbb{R}^2$ and $\mathbf{s} = (x,y)'$), but the problem can be simplified to one dimension (i.e., $D_s \in \mathbb{R}^1$ and $\mathbf{s} = x$) or generalized to higher spatial dimensions (i.e., $D_s \in \mathbb{R}^3$ and $\mathbf{s} = (x,y,z)'$). Finally, as is common in the dynamic systems literature, we refer to the N-dimensional multivariate vector $\mathbf{u}(\mathbf{s},t)$ as the state or system state, and use the term *component* to refer to each of the N elements of $\mathbf{u}(\mathbf{s},t)$.

We reparameterize (3.1) to be intrinsically linear (in parameters) as

$$\mathbf{u}_{t^{(J)}}(\mathbf{s},t) = \mathbf{M}\mathbf{f}(\mathbf{u}(\mathbf{s},t), \mathbf{u}_x(\mathbf{s},t), \mathbf{u}_y(\mathbf{s},t), \mathbf{u}_{xy}(\mathbf{s},t), \ldots, \mathbf{u}_{t^{(1)}}(\mathbf{s},t), \ldots, \mathbf{u}_{t^{(J-1)}}(\mathbf{s},t), \boldsymbol{\omega}(\mathbf{s},t)), \tag{3.2}$$

where $\mathbf{M}$ is a *sparse* matrix of coefficients of dimension $N \times D$ and $\mathbf{f}(\cdot)$ is a vector-valued nonlinear transformation function of length D. The input of the arguments for $\mathbf{f}(\cdot)$ are general and contain anything that *potentially* relates to the system. For example, this could include terms describing advection, diffusion, dispersion and growth, polynomial functions and interactions, or sinusoidal functions, and are chosen based on

a general mechanistic understanding of the system. This results in D being quite large, meaning (3.2) has the potential to be highly over-parameterized. Thus, we will employ regularization to induce sparsity in the matrix $\mathbf{M}$.

As an example of a classic PDE within our framework, consider the reaction-diffusion equation

$$\mathbf{u}_t(\mathbf{s}, t) = \mathbf{W}\nabla^2 \mathbf{u}(\mathbf{s}, t) + \mathbf{c}(\mathbf{u}(\mathbf{s}, t)),$$

where $\mathbf{u}(\mathbf{s}, t) = [b(\mathbf{s}, t), d(\mathbf{s}, t)]'$ represents the densities of two processes (i.e., $N = 2$), $\mathbf{W}$ is a diagonal matrix where $\operatorname{diag}(\mathbf{W}) = [W_b, W_d]$ are the diffusion constants, and $\mathbf{c}(\mathbf{u}(\mathbf{s}, t)) = [c_b(b(\mathbf{s}, t), d(\mathbf{s}, t)), c_d(b(\mathbf{s}, t), d(\mathbf{s}, t))]'$ are (non)linear reaction functions. The reaction-diffusion equation can be used to model the densities of prey (b) and predator (d) populations (Hastings, 1996). For a predator-prey model, a possible choice for the reaction function, with the spatial and temporal indices suppressed, is $\mathbf{c}(\mathbf{u}(\mathbf{s}, t)) = [\gamma b - \delta bd, -\mu d + \eta bd]'$. This is a simplistic representation of the Lotka-Volterra system where γ and δ represent the prey's birth and predation rates, respectively, and μ and η represent the predator death and kill success rates, respectively. Following (3.2), and again suppressing the spatial and temporal indices, we have

$$\begin{bmatrix} b_t \\ d_t \end{bmatrix} = \begin{bmatrix} \gamma & 0 & -\delta & W_b & W_b & 0 & 0 \\ 0 & -\mu & \eta & 0 & 0 & W_d & W_d \end{bmatrix} \begin{bmatrix} b & d & bd & b_{xx} & b_{yy} & d_{xx} & d_{yy} \end{bmatrix}'.$$

Typically we do not know $\mathbf{f}(\cdot) = [b, d, bd, b_{xx}, b_{yy}, d_{xx}, d_{yy}]'$ and instead highly over-parameterize $\mathbf{f}(\cdot)$ by including a library of potential terms and select against the coefficients in $\mathbf{M}$ to identify relevant terms.

In real-world problems, (3.2) does not hold exactly. Stochastic forcing could perturb the system (e.g., weather systems, demographic stochasticity) or there could be error in the model specification. We accommodate this unknown stochasticity by including an additive error term

$$\mathbf{u}_{t^{(J)}}(\mathbf{s}, t)$$
$$= \mathbf{M}\mathbf{f}\big(\mathbf{u}(\mathbf{s}, t), \mathbf{u}_x(\mathbf{s}, t), \mathbf{u}_y(\mathbf{s}, t), \mathbf{u}_{xy}(\mathbf{s}, t), \ldots, \mathbf{u}_{t^{(1)}}(\mathbf{s}, t), \ldots, \mathbf{u}_{t^{(J-1)}}(\mathbf{s}, t), \boldsymbol{\omega}(\mathbf{s}, t)\big) + \widetilde{\boldsymbol{\eta}}(\mathbf{s}, t),$$
$$(3.3)$$

where, for example, $\widetilde{\boldsymbol{\eta}}(\mathbf{s}, t) \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \widetilde{\boldsymbol{\Sigma}}_U)$ is a mean zero Gaussian process with variance/covariance matrix $\widetilde{\boldsymbol{\Sigma}}_U$. In general, spatial or temporal dependencies could be considered in this error term.

To represent (3.3) using tensor notation, let $\mathcal{U} = \{u(\mathbf{s}, t, n) : \mathbf{s} \in D_s, t = 1, \ldots, T, n = 1, \ldots, N\}$ where $\mathcal{U} \in \mathbb{R}^{S \times T \times N}$ is the tensor of the dynamic process. Similarly, let $\mathcal{F} \in \mathbb{R}^{S \times T \times D}$ be the function $\mathbf{f}(\cdot)$ evaluated at each location in space-time and $\widetilde{\boldsymbol{\eta}} \in \mathbb{R}^{S \times T \times N}$ is the space-time-component uncertainty tensor. The tensor formulation of (3.3) is then

$$\mathcal{U}_{t^{(J)}} = \mathcal{F} \times_3 \mathbf{M} + \widetilde{\boldsymbol{\eta}}. \tag{3.4}$$

This forms the core of our process model, where we relate the temporal derivative of some space-time-component process to a nonlinear function of its current state. While not explicitly stated in (3.4), $\mathcal{F}$ is still a function of the state process $\mathcal{U}$.

Continuing the reaction-diffusion example and placing it within the context of tensor notation, $\mathcal{U}_t$ is the $S \times T \times 2$ tensor of temporal derivatives where $(\mathbf{U}_{::1})_t$ and $(\mathbf{U}_{::2})_t$ are the first order temporal derivatives associated with prey and predator, respectively. Similarly, $\mathcal{F}$ is the $S \times T \times 7$ tensor of the function $f(\cdot)$ evaluated at all space time location. Each third-order slice of $\mathcal{F}$ represents the evaluated function at each space-time location. Specifically, for the reaction-diffusion example presented above, $\mathbf{F}_{::1}$ is the density of the prey population at each space-time location, $\mathbf{F}_{::2}$ is the density of the predator population at each space-time location, $\mathbf{F}_{::3}$ is the density of the product of the prey and predator population at each space-time location, $\mathbf{F}_{::4}$ is the second order derivative in the x direction of the predator population at each space-time location, and so on.

3.2   Basis Representation

As described in Section 2.2, we can represent the $\mathcal{U}$ tensor using basis functions. Decomposing $\mathcal{U}$ in terms of a finite collection of spatial, temporal, and component basis functions, we write

$$\mathcal{U} \approx \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} a(p,q,r)\boldsymbol{\psi}(p) \circ \boldsymbol{\phi}(q) \circ \boldsymbol{\theta}(r) = \mathcal{A} \times_1 \boldsymbol{\Psi} \times_2 \boldsymbol{\Phi} \times_3 \boldsymbol{\Theta} \coloneqq [\![\mathcal{A}; \boldsymbol{\Psi}, \boldsymbol{\Phi}, \boldsymbol{\Theta}]\!],$$

where $\mathcal{A} \in \mathbb{R}^{P \times Q \times R}$, $\boldsymbol{\Psi} \in \mathbb{R}^{S \times P}$, $\boldsymbol{\Phi} \in \mathbb{R}^{T \times Q}$, and $\boldsymbol{\Theta} \in \mathbb{R}^{N \times R}$. Here, $\boldsymbol{\Psi}, \boldsymbol{\Phi}$, and $\boldsymbol{\Theta}$ are matrices of spatial, temporal, and component basis functions, respectively, and $\mathcal{A}$ is a tensor of basis coefficients (traditionally called the *core tensor*).

We can obtain derivatives of the elements of $\mathcal{U}$ analytically by taking derivatives of the basis functions. Specifically, let $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ be matrices of basis functions differentiable up to at least the highest order considered in (3.1). We then compute spatial and temporal derivatives of $\mathcal{U}$ by computing the derivatives of $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$. That is, denote $\frac{\partial}{\partial x}\boldsymbol{\Psi} = \boldsymbol{\Psi}_x$, $\frac{\partial}{\partial y}\boldsymbol{\Psi} = \boldsymbol{\Psi}_y$, $\frac{\partial}{\partial t}\boldsymbol{\Phi} = \boldsymbol{\Phi}_t$, and so forth. Derivatives of $\mathcal{U}$ are then computed as

$$\begin{aligned}
\frac{\partial}{\partial t}\mathcal{U} &= \mathcal{A} \times_1 \boldsymbol{\Psi} \times_2 \boldsymbol{\Phi}_t \times_3 \boldsymbol{\Theta} = [\![\mathcal{A}; \boldsymbol{\Psi}, \boldsymbol{\Phi}_t, \boldsymbol{\Theta}]\!], \\
\frac{\partial}{\partial x}\mathcal{U} &= \mathcal{A} \times_1 \boldsymbol{\Psi}_x \times_2 \boldsymbol{\Phi} \times_3 \boldsymbol{\Theta} = [\![\mathcal{A}; \boldsymbol{\Psi}_x, \boldsymbol{\Phi}, \boldsymbol{\Theta}]\!], \\
\frac{\partial^2}{\partial x \partial y}\mathcal{U} &= \mathcal{A} \times_1 \boldsymbol{\Psi}_{xy} \times_2 \boldsymbol{\Phi} \times_3 \boldsymbol{\Theta} = [\![\mathcal{A}; \boldsymbol{\Psi}_{xy}, \boldsymbol{\Phi}, \boldsymbol{\Theta}]\!],
\end{aligned} \tag{3.5}$$

and so forth. Representing (3.4) using the basis decomposition, we have

$$[\![\mathcal{A}; \boldsymbol{\Psi}, \boldsymbol{\Phi}_{t^{(J)}}, \boldsymbol{\Theta}]\!] = \mathcal{F} \times_3 \mathbf{M} + \boldsymbol{\eta},$$

where $\boldsymbol{\eta}$ may include truncation error due to the low-rank representation of the basis functions. While not explicitly stated, $\mathcal{F}$ now depends on $\boldsymbol{\Psi}, \boldsymbol{\Phi}, \boldsymbol{\Theta}$, and $\mathcal{A}$.

Proposition 1. *The mode-3 decomposition of* $[\![\mathcal{A}; \boldsymbol{\Psi}, \boldsymbol{\Phi}_{t^{(J)}}, \boldsymbol{\Theta}]\!] = \mathcal{F} \times_3 \mathbf{M} + \boldsymbol{\eta}$ *where* $\boldsymbol{\eta}(\mathbf{s}, t) \overset{i.i.d.}{\sim} N_N(\mathbf{0}, \Sigma_U)$ *in space and time at location* $\mathbf{s}$ *and time* t *is*

$$
\begin{aligned}
&\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\phi}_{t^{(J)}}(t) \otimes \boldsymbol{\psi}(\mathbf{s}))' \\
&= \mathbf{Mf}(\mathbf{A}, \boldsymbol{\psi}(\mathbf{s}), \boldsymbol{\psi}_x(\mathbf{s}), \boldsymbol{\psi}_y(\mathbf{s}), \boldsymbol{\psi}_{xy}(\mathbf{s}), \ldots, \boldsymbol{\phi}_{t^{(0)}}(t), \ldots, \boldsymbol{\phi}_{t^{(J)}}(t), \boldsymbol{\omega}(\mathbf{s}, t)) + \boldsymbol{\eta}(\mathbf{s}, t),
\end{aligned}
$$

where $\otimes$ *denotes the Kronecker product,* $\mathbf{A}$ *is a* $R \times PQ$ *matrix of basis coefficients,* $\boldsymbol{\psi}(\mathbf{s})$ *is a length-P vector of spatial basis functions,* $\boldsymbol{\phi}(t)$ *is a length-Q vector of temporal basis functions, and* $\boldsymbol{\Theta}$ *is a* $N \times R$ *matrix of component basis functions.*

Proof. See Supplementary Material (North et al., 2023). □

Decomposing (3.4) in terms of basis functions and taking the mode-3 matricization accomplishes two tasks. First, this enables inference on derivatives of the process $\mathbf{u}(\mathbf{s}, t)$ when only the process is known (e.g., see (3.5)). Second, keeping fewer basis functions than observations (e.g., $P < S$, $Q < T$) allows the reconstruction of $\mathcal{U}$ to be smooth (Wang et al., 2016).

Note, we include $\boldsymbol{\Theta}$ for generality in the construction of our method. While one could specify $\boldsymbol{\Theta}$ in terms of basis functions, our goal is not to reduce the dimension of the system state variables. In our analyses, we choose $\boldsymbol{\Theta}$ to be the identity matrix.

3.3 Transformation of Derivative

Up to this point, we have considered PDEs that relate the temporal derivative (of some order J) of the continuous surface $\mathbf{u}$ on left-hand side (LHS) of (3.1) to a function of its current state on the right-hand side (RHS) of (3.1). However, equations with a spatio-temporal derivative of $\mathbf{u}$ on the LHS are common (e.g., vorticity equation, Higham et al., 2016). For example, the LHS of (3.1) could depend on the Laplacian operator, where $\nabla^2 \mathbf{u}_{t^{(J)}}(\mathbf{s}, t) = \mathbf{u}_{xxt^{(J)}}(\mathbf{s}, t) + \mathbf{u}_{yyt^{(J)}}(\mathbf{s}, t)$.

To be more general, we now allow the LHS of (3.1) to be a function of spatio-temporal derivatives of $\mathbf{u}$ and consider the more general PDE

$$
g(\mathbf{u}_{t^{(J)}}(\mathbf{s}, t)) = M(\mathbf{u}(\mathbf{s}, t), \mathbf{u}_x(\mathbf{s}, t), \mathbf{u}_y(\mathbf{s}, t), \mathbf{u}_{xy}(\mathbf{s}), \ldots, \mathbf{u}_{t^{(1)}}(\mathbf{s}, t), \ldots, \mathbf{u}_{t^{(J-1)}}(\mathbf{s}, t), \boldsymbol{\omega}(\mathbf{s}, t)),
\tag{3.6}
$$

where $g(\cdot)$ is some linear differential operator. The original PDE (3.1) is a special case of (3.6) where $g(\cdot)$ is the identity function.

Proposition 2. *Let* $g(\cdot)$ *be a linear differential operator. The basis formulation of a PDE with a space-time function* $g(\mathbf{u}_{t^{(J)}}(\mathbf{s}, t))$ *on the LHS is*

$$
\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\phi}_{t^{(J)}}(t) \otimes g(\boldsymbol{\psi}(\mathbf{s})))'.
$$

Proof. See Supplementary Material (North et al., 2023). □

From Proposition 2, the basis representation of a PDE with a spatio-temporal function on the LHS is

$$\mathbf{\Theta}\mathbf{A}(\boldsymbol{\phi}_{t^{(J)}}(t) \otimes g(\boldsymbol{\psi}(\mathbf{s})))'$$
$$= \mathbf{Mf}(\mathbf{A}, \boldsymbol{\psi}(\mathbf{s}), \boldsymbol{\psi}_x(\mathbf{s}), \boldsymbol{\psi}_y(\mathbf{s}), \boldsymbol{\psi}_{xy}(\mathbf{s}), \dots, \boldsymbol{\phi}_{t^{(0)}}(t), \dots, \boldsymbol{\phi}_{t^{(J)}}(t), \boldsymbol{\omega}(\mathbf{s}, t)) + \boldsymbol{\eta}(\mathbf{s}, t), \tag{3.7}$$

where $\boldsymbol{\eta}(\mathbf{s}, t) \overset{i.i.d.}{\sim} N_N(\mathbf{0}, \Sigma_U)$ in space and time. Completing the example from before using the $g = \nabla^2$ Laplacian operator, the LHS for (3.7) is $\mathbf{\Theta}\mathbf{A}(\boldsymbol{\phi}_{t^{(J)}}(t) \otimes (\boldsymbol{\psi}_{xx}(\mathbf{s}) + \boldsymbol{\psi}_{yy}(\mathbf{s})))'$.

3.4 Data Model

We assume $\mathbf{v}(\mathbf{s}, t)$ is an observation of the N-dimensional latent process $\mathbf{u}(\mathbf{s}, t)$ outlined in Section 3.2 with some unknown measurement uncertainty. We model $\mathbf{v}(\mathbf{s}, t)$ using a generalization to the traditional linear measurement error model that links the dynamics to the observed process (e.g., see Cressie and Wikle, 2011, Chapter 7). That is, we model

$$\mathbf{v}(\mathbf{s}, t) = \mathbf{H}(\mathbf{s}, t)\mathbf{u}(\mathbf{s}, t) + \widetilde{\boldsymbol{\epsilon}}(\mathbf{s}, t), \tag{3.8}$$

where $\mathbf{v}(\mathbf{s}, t) \in \mathbb{R}^{L(\mathbf{s}, t)}$, $\mathbf{H}(\mathbf{s}, t) \in \mathbb{R}^{L(\mathbf{s}, t) \times N}$ is the incidence matrix that maps from $\mathbf{u}(\mathbf{s}, t)$ to $\mathbf{v}(\mathbf{s}, t)$, and uncertainty in the observations of the process are captured by $\widetilde{\boldsymbol{\epsilon}}(\mathbf{s}, t) \overset{\text{indep.}}{\sim} N_{L(\mathbf{s}, t)}(\mathbf{0}, \widetilde{\mathbf{\Sigma}}_V(\mathbf{s}, t))$. The dimension of the data, $L(\mathbf{s}, t) \leq N$, is allowed to vary based on the space-time location due to potentially missing data and we assume the errors are independent in space and time.

Within the hierarchical model, missing data are accommodated by allowing the dimension of the incidence matrix, $\mathbf{H}(\mathbf{s}, t)$, to vary in time. Since missing data are handled in the data model and the latent process is fully specified, missing data do not impact the process model specification. If there are no missing data at time t and location $\mathbf{s}$, then $L(\mathbf{s}, t) = N$ and $\mathbf{H}(\mathbf{s}, t) = \mathbf{I}_N$. When one or more system components are missing data, the row corresponding to the missing system component is removed. For example, if we have a three-dimensional system, say $\mathbf{u}(\mathbf{s}, t) = [a(\mathbf{s}, t), b(\mathbf{s}, t), c(\mathbf{s}, t)]$ and the observation component for $b(\mathbf{s}, t)$ is missing at location $\mathbf{s}$ and time t, then

$$\mathbf{H}(\mathbf{s}, t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

See Chapter 7 of Cressie and Wikle (2011) for more discussion of this approach for accommodating missing observations in hierarchical spatio-temporal models.

Incorporating the basis expansion of the process in (3.8), at location $\mathbf{s}$ and time t,

$$\mathbf{v}(\mathbf{s}, t) = \mathbf{H}(\mathbf{s}, t)\mathbf{\Theta}\mathbf{A}(\boldsymbol{\phi}_{t^{(0)}}(t) \otimes \boldsymbol{\psi}(\mathbf{s}))' + \boldsymbol{\epsilon}(\mathbf{s}, t), \tag{3.9}$$

where $\boldsymbol{\epsilon}(\mathbf{s}, t) \overset{indep.}{\sim} N_L(\mathbf{0}, \Sigma_V(\mathbf{s}, t))$ and $\boldsymbol{\epsilon}(\mathbf{s}, t)$ now accounts for both measurement error and the discrepancy between the "true" underlying process and our approximation using the basis formulation.

3.5 Parameter Model

The data and process equations correspond to the first and second level of our hierarchical model, respectively. For convenience, we restate (3.9) and (3.7) for location $\mathbf{s}$ and time t

$$\mathbf{v}(\mathbf{s}, t) = \mathbf{H}(\mathbf{s}, t)\mathbf{\Theta}\mathbf{A}(\boldsymbol{\phi}_{t^{(0)}}(t) \otimes \boldsymbol{\psi}(\mathbf{s}))' + \boldsymbol{\epsilon}(\mathbf{s}, t),$$

$$\mathbf{\Theta}\mathbf{A}(\boldsymbol{\phi}_{t^{(J)}}(t) \otimes g(\boldsymbol{\psi}(\mathbf{s})))'$$
$$= \mathbf{M}\mathbf{f}(\mathbf{A}, \boldsymbol{\psi}(\mathbf{s}), \boldsymbol{\psi}_x(\mathbf{s}), \boldsymbol{\psi}_y(\mathbf{s}), \boldsymbol{\psi}_{xy}(\mathbf{s}), \ldots, \boldsymbol{\phi}_{t^{(0)}}(t), \ldots, \boldsymbol{\phi}_{t^{(J)}}(t), \boldsymbol{\omega}(\mathbf{s}, t)) + \boldsymbol{\eta}(\mathbf{s}, t),$$

where $\boldsymbol{\epsilon}(\mathbf{s}, t) \stackrel{indep.}{\sim} N_{L(\mathbf{s},t)}(\mathbf{0}, \boldsymbol{\Sigma}_V(\mathbf{s}, t))$ and $\boldsymbol{\eta}(\mathbf{s}, t) \stackrel{i.i.d.}{\sim} N_N(\mathbf{0}, \boldsymbol{\Sigma}_U)$. For clarity, we present the details on the model parameters in Table 1. Our goal is to make inference on the unknown parameters $\mathbf{M}, \boldsymbol{\Sigma}_U, \boldsymbol{\Sigma}_V(\mathbf{s}, t)$, and $\mathbf{A}$. The sparse matrix $\mathbf{M}$ identifies the nonlinear dynamic equation, $\boldsymbol{\Sigma}_U$ captures the error dependencies within the dynamic equation, $\boldsymbol{\Sigma}_V(\mathbf{s}, t)$ captures the measurement uncertainty associated with the observed data, and $\mathbf{A}$ defines the smooth latent process.

To complete our Bayesian hierarchical model, we define the following priors on these parameters. We use the spike-and-slab prior (Mitchell and Beauchamp, 1988; George et al., 1993) to induce sparsity into $\mathbf{M}$. We write

$$\mathbf{M}(n, \cdot)|\mathbf{\Gamma}(n, \cdot), \boldsymbol{\sigma}_U^2(n) = \prod_{d=1}^{D}[(1 - \gamma(n, d))\delta_0 + \gamma(n, d)p(m(n, d)|\sigma_U^2(n), \cdot)],$$

where $\mathbf{\Gamma}$ is a matrix of inclusion indicators of the same dimension as $\mathbf{M}$, $\mathbf{M}(n, \cdot)$ and $\mathbf{\Gamma}(n, \cdot)$ are the nth row of the $\mathbf{M}$ and $\mathbf{\Gamma}$, respectfully, $m(n, d)$ and $\gamma(n, d)$ denote the dth column and nth row of $\mathbf{M}$ and $\mathbf{\Gamma}$, respectfully, δ_0 denotes the Dirac function at 0, $\boldsymbol{\sigma}_U^2(n)$ is the nth diagonal component of $\boldsymbol{\Sigma}_U$, $p(\gamma(n, d) = 1|\pi_n) = \pi(n)$, and $\pi(n) \sim Beta(a, b)$. That is, if a variable is not included (i.e., $\gamma(n, d) = 0$), then the corresponding element $m(n, d)$ is zero. If a variable is included (i.e., $\gamma(n, d) = 1$), then the corresponding element $m(n, d)$ is non-zero. There are multiple choices for the prior $p(m(n, d)|\sigma_U^2(n), \cdot)$. We specify the g-slab prior corresponding to Zellner's g-prior (Zellner, 1986) where scaling factor is taken to be the number of observations. See Malsiner-Walli and Wagner (2016) for other potential choices and further discussion.

While other shrinkage/selection priors could be used, such as Stochastic Search Variable Selection (SSVS; George et al., 1993), LASSO (Park and Casella, 2008), or Horseshoe (Carvalho et al., 2010), we found the spike-and-slab to be preferable since it performs well with correlated predictors (Ročková and George, 2014), which are generally present in the feature library (see Section 4). Additionally, the posterior summary of the latent variable $\boldsymbol{\gamma}(n, d)$ gives the inclusion probability for each component of $\mathbf{M}$, providing further insight into the certainty of the recovered system. For all examples presented below, we determine the identified system as composed of terms that are included with at least 50% posterior probability. However, this threshold is subjective and one could choose a different value depending on their specific application.

To estimate $\mathbf{\Gamma}$ and avoid reducibility of the Markov chain, we compute the marginal posterior distribution $[\mathbf{\Gamma}|\mathbf{A}] \propto [\mathbf{A}|\mathbf{\Gamma}][\mathbf{\Gamma}]$. To compute this conditional distribution, we

Model	Symbol	Description	Dimension
	Variable		
Data	$\mathbf{v}(\mathbf{s}, t)$	Observed data	$L(\mathbf{s}, t) \times 1$
Data	$\mathbf{H}(\mathbf{s}, t)$	Mapping matrix	$L(\mathbf{s}, t) \times N$
Data	$\boldsymbol{\epsilon}(\mathbf{s}, t)$	Data uncertainty vector	$L(\mathbf{s}, t) \times 1$
Data	$\boldsymbol{\Sigma}_V(\mathbf{s}, t)$	Measurement error covariance matrix	$L(\mathbf{s}, t) \times L(\mathbf{s}, t)$
Process	$\mathbf{u}(\mathbf{s}, t)$	Dynamic process	$N \times 1$
Process	$\mathcal{A}$	Basis coefficient tensor	$P \times Q \times N$
Process	$\mathbf{A}$	Basis coefficient matrix (mode-3)	$N \times (P \times Q)$
Process	$\boldsymbol{\psi}(\mathbf{s})$	spatial basis function for location $\mathbf{s}$	$P \times 1$
Process	$\boldsymbol{\phi}_{t(j)}(t)$	jth order temporal basis function for time t	$Q \times 1$
Process	$\boldsymbol{\Theta}$	Component basis function matrix	$N \times R$
Process	$\mathbf{M}$	Dynamic evolution matrix	$N \times D$
Process	$\boldsymbol{\Gamma}$	Inclusion indicator matrix	$N \times D$
Process	$\mathbf{f}(\cdot)$	Feature library	$D \times 1$
Process	$\boldsymbol{\eta}(\mathbf{s}, t)$	Process uncertainty vector	$N \times 1$
Process	$\boldsymbol{\Sigma}_U$	Dynamic equation error covariance matrix	$N \times N$
	Dimension		
	T	Number of observed time points	1
	S	Number of observed spatial locations	1
	$L(\mathbf{s}, t)$	Dimension of observation vector at time t and location $\mathbf{s}$	1
	N	Dimension of latent process (dynamic system)	1
	D	Number of library functions	1
	P	Number of spatial basis functions	1
	Q	Number of temporal basis functions	1
	R	Number of component basis functions	1
	J	Highest order derivative in the dynamic system	1
	Indices		
	t	Time interval, $t \in \{1, \ldots, T\} = D_t$, $|D_t| = T$	1
	$\mathbf{s}$	Spatial location, $\mathbf{s} \in \{\mathbf{s}_1, \ldots, \mathbf{s}_S\} = D_s$, $|D_s| = S$	1
	j	Order of the derivative, $j = 1, \ldots, J$	1

Table 1: List of symbols used in the Bayesian hierarchical model.

need to compute $[\mathbf{A}|\boldsymbol{\Gamma}]$, which is obtained by integrating over the parameters subject to selection:

$$[\mathbf{A}|\boldsymbol{\Gamma}] = \int \int [\mathbf{A}|\mathbf{M}, \boldsymbol{\Sigma}_U, \boldsymbol{\Gamma}][\mathbf{M}][\boldsymbol{\Sigma}_U] d\mathbf{M} d\boldsymbol{\Sigma}_U. \tag{3.10}$$

To make the integration in (3.10) analytically tractable and keep conjugacy in the model, we restrict $\boldsymbol{\Sigma}_U$ to be diagonally structured where $\boldsymbol{\Sigma}_U = \text{diag}(\sigma_U^2(1), \ldots, \sigma_U^2(N))$ and each diagonal element is assigned the non-informative prior $\sigma_U^2(n) \propto 1/\sigma_U^2(n), n = 1, \ldots, N$. Then, the probability any element is included is given as

$$p(\gamma(n, d) = 1|\cdot) = \frac{1}{1 + \frac{1-\pi(n)}{\pi(n)} R_\gamma(n, d)}, \tag{3.11}$$

where

$$R_\gamma(n, d) = \frac{[\mathbf{A}|\gamma(n, d) = 0]}{[\mathbf{A}|\gamma(n, d) = 1]}.$$

In situations where dependence between the components is required, a different prior could be used.

There is potential for elements of the variance-covariance matrix $\mathbf{\Sigma}_V(\mathbf{s}, t)$ to have small values. Inference using traditional conjugate Inverse Gamma/Wishart priors are overly sensitive to the choice of hyperpriors when estimates are small (Gelman, 2006). Instead, we use the conjugate Half-t prior proposed by Huang and Wand (2013) for covariance estimation, which imposes less prior information and does not have as strong of influence on small estimates. We restrict the measurement error covariance matrix to be diagonally structured since it is often a reasonable assumption that this is independent (Cressie and Wikle, 2011) (although this restriction can be removed if warranted). Let $\mathbf{\Sigma}_V(\mathbf{s}, t) = \mathbf{H}(\mathbf{s}, t) \operatorname{diag}(\sigma_V^2(1), \ldots, \sigma_V^2(N))\mathbf{H}(\mathbf{s}, t)'$, where each diagonal element, $\sigma_V(1), \ldots, \sigma_V(N)$, is assigned a conjugate Half-t$(2, 10^{-5})$ prior.

Finally, in order to induce sparcity in the basis coefficients, we assign a Bayesian elastic net prior (Li and Lin, 2010) to $\mathbf{A}$. Specifically, our prior is

$$\pi(\mathbf{A}) \propto \exp\{-\lambda_1 \|\mathbf{A}\|_1 - \lambda_2 \|\mathbf{A}\|_2^2\},$$

where λ_1, λ_2 are penalty parameters. The elastic net prior helps regularize the basis coefficients against basis functions. While it is possible to specify hyperpriors for the two penalty terms, we find inference is not overly sensitive to the choice of penalty parameters and fix them each to a small value (e.g., 0.01 or 0.001).

4 Model Estimation

Our goal is to obtain samples from the joint posterior distribution $[\mathbf{M}, \mathbf{\Sigma}_U, \mathbf{\Sigma}_V, \mathbf{\Gamma}, \mathbf{A}|\cdot]$. We achieve this by sampling from the five full-conditional distributions $[\mathbf{M}|\cdot]$, $[\mathbf{\Sigma}_U|\cdot]$, $[\mathbf{\Sigma}_V|\cdot]$, $[\mathbf{\Gamma}|\cdot]$, and $[\mathbf{A}|\cdot]$ (see the Supplementary Material for the details of the distributions and sampling algorithm) using a Markov chain Monte Carlo (MCMC) sampling scheme. The four components $\mathbf{M}, \mathbf{\Sigma}_U, \mathbf{\Sigma}_V$ and $\mathbf{\Gamma}$ are updated using classical Bayesian methods and $\mathbf{A}$ is updated using a stochastic gradient approach. Due to the variety of problems for which our method is applicable, some modeling choices are case specific. Additionally, some aspects of the implementation of the MCMC framework warrant a more detailed discussion. The following sections provide additional information pertaining to these model specifications and procedures.

4.1 Basis Coefficient Estimation

The basis coefficients, $\mathbf{A}$, completely define the latent process and all derivatives in both space and time, meaning proper estimation is crucial to the discovery process. Since $\mathbf{A}$ is embedded within the nonlinear function $\mathbf{f}(\cdot)$ (see Proposition 1) and $\mathbf{f}(\cdot)$ is problem specific, we need a method for estimating $\mathbf{A}$ that can be generalized to a different systems. To accommodate a generically specified $\mathbf{f}(\cdot)$, we use an adapted version of stochastic gradient descent (SGD) with a constant learning rate (SGDCL; Mandt et al., 2016). Whereas other approaches to estimate $\mathbf{A}$ (e.g., Expectation-Maximization

or Metropolis-Hastings) could be used, SGDCL provides important advantages – a conjugate updating scheme and a reduced computational cost for any specification of $\mathbf{f}(\cdot)$.

As with SGD, SGDCL relies on the gradient of a loss function and a learning rate. For SGDCL, the loss function is the negative log posterior for our parameters of interest, $\mathbf{A}$. The loss function at location $\mathbf{s}$ and time t is

$$
\begin{aligned}
&\mathcal{L}(\mathbf{A}; \mathbf{s}, t) \\
&= -\log([\mathbf{v}(\mathbf{s}, t)|\mathbf{A}, \mathbf{H}(\mathbf{s}, t), \boldsymbol{\Theta}, \boldsymbol{\phi}_{t^{(0)}}(t), \boldsymbol{\psi}(\mathbf{s}), \boldsymbol{\Sigma}_V] \\
&\qquad [\mathbf{A}, \boldsymbol{\Theta}, \boldsymbol{\phi}_{t^{(i)}}(t), g(\boldsymbol{\psi}(s))|\mathbf{M}, \boldsymbol{\Sigma}_U, \mathbf{A}, \boldsymbol{\Theta}, \boldsymbol{\phi}_{t^{(0)}}(t), \ldots, \boldsymbol{\phi}_{t^{(J-1)}}(t), \boldsymbol{\psi}(\mathbf{s}), \boldsymbol{\psi}_x(\mathbf{s}), \ldots]) \\
&\quad -\log([\mathbf{A}]).
\end{aligned}
\tag{4.1}
$$

The deterministic matrices $\boldsymbol{\Theta}, \boldsymbol{\phi}$, and $\boldsymbol{\psi}$ are included in the LHS of the conditioning statement in (4.1) to make it clear which process, or derivative, is being considered. To simplify notation, denote $\mathbf{B}_0(\mathbf{s}, t) = \boldsymbol{\phi}_{t^{(0)}}(t) \otimes \boldsymbol{\psi}(\mathbf{s})$ and $\mathbf{B}_J(\mathbf{s}, t) = \boldsymbol{\phi}_{t^{(J)}}(t) \otimes g(\boldsymbol{\psi}(\mathbf{s}))$. Then, the gradient of the loss function $\mathcal{L}(\mathbf{A}; \mathbf{s}, t)$, $\frac{\partial \mathcal{L}(\mathbf{A}; \mathbf{s}, t)}{\partial \mathbf{A}} = \nabla_{\mathbf{A}} \mathcal{L}(\mathbf{A}; \mathbf{s}, t)$ for location $\mathbf{s}$ and time t is

$$
\begin{aligned}
\nabla \mathcal{L}(\mathbf{A}; \mathbf{s}, t) = &- \boldsymbol{\Theta}' \mathbf{H}'(\mathbf{s}, t) \boldsymbol{\Sigma}_V^{-1} \mathbf{v}(\mathbf{s}, t) B_0(\mathbf{s}, t) + \boldsymbol{\Theta}' \mathbf{H}'(\mathbf{s}, t) \boldsymbol{\Sigma}_V^{-1} \mathbf{H}(\mathbf{s}, t) \boldsymbol{\Theta} \mathbf{A} \mathbf{B}_0'(\mathbf{s}, t) \mathbf{B}_0(\mathbf{s}, t) \\
&+ \boldsymbol{\Theta}' \boldsymbol{\Sigma}_U^{-1} \boldsymbol{\Theta} \mathbf{A} \mathbf{B}_J'(\mathbf{s}, t) \mathbf{B}_J(\mathbf{s}, t) - \boldsymbol{\Theta}' \boldsymbol{\Sigma}_U^{-1} \mathbf{M} \mathbf{f}(\mathbf{s}, t) \mathbf{B}_J(\mathbf{s}, t) \\
&- \mathbf{B}_J(\mathbf{s}, t) \mathbf{A}' \boldsymbol{\Theta}' \boldsymbol{\Sigma}_U^{-1} \mathbf{M} \dot{\mathbf{F}}'(\mathbf{s}, t) + \mathbf{f}'(\mathbf{s}, t) \mathbf{M}' \boldsymbol{\Sigma}_U^{-1} \mathbf{M} \dot{\mathbf{F}}'(\mathbf{s}, t) \\
&+ \frac{1}{ST} \left(\lambda_1 \operatorname{sign}(\mathbf{A}) + 2\lambda_2 \mathbf{A} \right),
\end{aligned}
$$

where $\dot{\mathbf{F}}(\mathbf{s}, t)$ generically denotes $\frac{\partial}{\partial \mathbf{A}} \mathbf{f}(\mathbf{A}, \cdot)$.

SGDCL (Mandt et al., 2016) replaces the true gradient with the stochastic estimate,

$$
\widehat{\nabla \mathcal{L}}_{\mathcal{Z}}(\mathbf{A}) = \frac{1}{|\mathcal{Z}|} \sum_{z \in \mathcal{Z}} \nabla_{\mathbf{A}} \mathcal{L}(\mathbf{A}; z),
$$

where $\mathcal{Z} \subset D_s \times D_t$ is a random subset of the observations, called a mini-batch, and $|\mathcal{Z}|$ is the cardinality of the set. Within the context of a MCMC algorithm, the ℓth update of $\mathbf{A}$ is given by

$$
\mathbf{A}^{(\ell)} = \mathbf{A}^{(\ell-1)} - \kappa' \widehat{\nabla \mathcal{L}}_{\mathcal{Z}^{(\ell)}}(\mathbf{A}^{(\ell-1)}),
$$

where $\mathcal{Z}^{(\ell)}$ denotes a random minibatch specific to the ℓ update and κ is the learning rate. To accommodate different scales for each component, we allow κ to be a vector of length N where each component can have a specific learning rate.

The final challenge to estimating $\mathbf{A}$ is computing $\dot{\mathbf{F}}(\mathbf{s}, t)$. Because $\mathbf{f}(\cdot)$ is problem specific, $\dot{\mathbf{F}}(\mathbf{s}, t)$ is also problem specific. One option is to use automatic differentiation (AD) to analytically compute the derivative of $\mathbf{f}(\cdot)$. There are many different libraries and programs that perform AD, and we explored the use of the *ForwardDiff* (Revels et al., 2016) package in Julia (Bezanson et al., 2017) with success. However, there is computational overhead to AD. So, for all the examples presented here we derive $\dot{\mathbf{F}}(\mathbf{s}, t)$ analytically to mitigate this computation bottleneck.

4.2   Choice of Basis Functions

The choice of basis functions are subjective and have the potential to affect the model fit (North et al., 2022). Furthermore, the choices of spatial and temporal basis functions do not need to be the same (e.g., radial basis functions in space and Fourier basis functions in time). There are other choices regarding basis functions that need to be taken into consideration (see Ramsay and Silverman, 2005, Chapter 3 for a discussion on how to choose basis functions based on the "shape" of the data). The most important requirement is that the spatial and temporal basis functions need to be differentiable up to at least the highest order spatial and temporal derivative considered, respectively. We found local basis functions (e.g., B-splines) perform better than global basis functions (e.g., Fourier basis functions) (see North et al., 2022), especially when there are local regions with minimal curvature. For these reasons, we use B-splines of an order greater than our highest derivative (in both space and time). We need to include enough basis functions such that the estimated solution curve is flexible, the dynamics are captured, and the posterior latent space is properly explored, but not so many such that unnecessary noise is introduced into the system. Empirically, we found a ratio of approximately 1 basis function to every 3 to 5 observations to work well. For example, in a system with 100 spatial locations and 200 temporal observations, we recommend letting $P = 20$ to 35 and $Q = 40$ to 65, respectively.

4.3   Choice of Feature Library

The choice of functions for the feature library is crucial to the identification of the system. Our method is restricted to search over a predefined set of functions, meaning that our method is unable to identify an important function if it is not included in the library. For this reason, it is best to over-parameterize the feature library (and hence $\mathbf{M}$) instead of specifying a restrictive set of functions. Additionally, some knowledge of the problem is beneficial (i.e., this is not a black-box approach). Having an understanding of the potential dynamics *a priori* can assist in the recovery of important dynamics. For example, if the system appears to diffuse over time, then a diffusion term should be included. A good default choice is to include polynomial terms that interact with varying orders of spatial and/or temporal derivatives of the process (e.g., see the library for Burgers' example) as this will cover a wide collection of systems.

With regard to the choice of $g()$ in (3.6) and (3.7), scientific knowledge of the problem is required. The choice of $g()$ is not searched over as with the library terms; rather it is pre-specified. For example, in our real-world example, the *a prior* goal was to make inference on the change of atmospheric vorticity with time, and vorticity can be represented as the Laplacian of the streamfunction. Because this transformation function is not learned, it is a modeling choice that is user specified (the default choice should be the identity as in Proposition 1).

4.4   Library Multicollinearity

A major issue facing the identification of spatio-temporal dynamic equations is multicollinearity in the feature library. Figure 1 shows the correlation between different components of a library using data generated from Burgers' equation. In this example, the polynomial terms, u, u^2, and u^3, are very positively linearly correlated ($\rho > 0.8$), posing a challenge to parameter inference. As with classical regression, multicollinearity has the potential to introduce bias into the coefficient estimates, including altering their sign. While the spike-and-slab has been shown to perform well with correlated variables, as discussed in Section 3.5, the problem still persists and can pose an estimation issue in problems similar to the example using Burgers' equation.
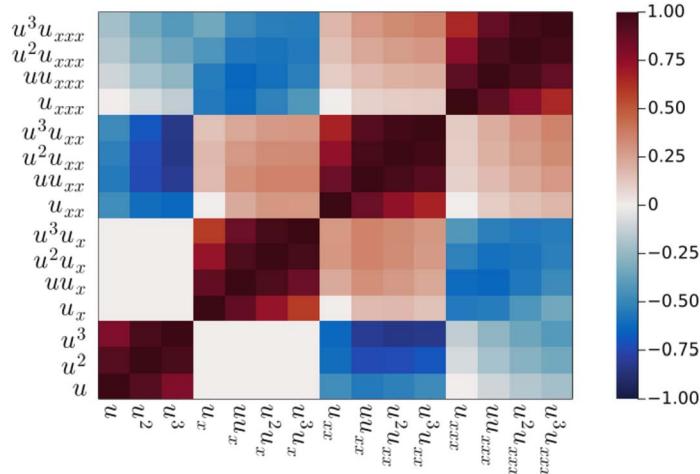
Figure 1: Correlation between terms of a potential feature library using data generated from Burgers' equation.

This issue originates from the over-inflation of minor reductions in the residual sum of squares (RSS) when a highly correlated, but incorrect term, is included. Specifically, the probability of including any term in the library (3.11) is dependent on the ratio of the residual sum of squares for the model with the $m(n, d)$ term included (RSS_γ) to the model without the $m(n, d)$ term included ($RSS_{\backslash\gamma}$) through the value of $R_\gamma(n, d)$. That is, under the *g*-prior

$$R_\gamma(n, d) = (g + 1)^{1/2} (RSS_\gamma / RSS_{\backslash\gamma})^{ST/2 - 1}.$$

Because $RSS_\gamma < RSS_{\backslash\gamma}$, the ratio is bound between 0 and 1, where correct terms in the library result in the ratio being closer to 0 and incorrect terms result in the ratio being close to 1. However, as the ratio is raised to a power of $ST/2 - 1$ (proportional to the number of observations), the value of R goes to 0 as the number of observations goes to infinity, resulting in all variables being found significant. This issue is exacerbated by correlated variables, especially if there are multiple confounding variables where the

system can be approximated by some linear combination of the feature library *without* the true terms being included.

To combat this issue we propose a method to reduce the impact of correlated variables (i.e., variables where in the ratio of RSSs being close to 1 are found significant). For this, we subsample the process when estimating the inclusion latent variable γ. That is, to compute (3.11) within each iteration of the Gibbs sampler, we randomly sample the process. This results in

$$R_\gamma^*(n, d) = (g + 1)^{1/2}(RSS_{\setminus\gamma}/RSS_\gamma)^{S^*T^*/2-1},$$

where S^* and T^* are the size of the subsampled dimensions. We provide details on how to choose the subsample size and our choices for the examples in the Supplementary Material. Note that this subsampling is only done for the $\boldsymbol{\Gamma}$ update step of the algorithm.

5 Simulations

We illustrate the ability of proposed model to discover dynamic equations using data simulated from two well known systems – Burgers' equation and a reaction-diffusion system. A third example, the heat equation, is included in the Supplementary Material. For all three examples, we investigate the impact of measurement error on inference. We introduce measurement error by adding white noise to the state vector. Specifically, we let $\mathbf{v}(\mathbf{s}, t) = \mathbf{u}(\mathbf{s}, t) + \zeta\boldsymbol{\epsilon}(\mathbf{s}, t)$, where $\mathbf{u}(\mathbf{s}, t)$ is the simulated data, $\boldsymbol{\epsilon}(\mathbf{s}, t) \sim N(\mathbf{0}, \sigma^2\mathbf{I}_N)$ is the additive noise, σ is the standard deviation of the simulated process $\mathbf{u}(\mathbf{s}, t)$, and ζ is the percent of noise ranging from 0 to 100 – henceforth referred to as the $\zeta\%$ *noise*. In addition, we show how the model performs when data are missing at random for Burger's equation. Unless otherwise stated, all reported estimates are rounded to three significant digits for readability. For all simulations and real-world examples, we obtain 5000 posterior samples and discard the first 2500 as burn-in. Convergence is assessed visually via trace plots with no issues detected.

5.1 Burgers' Equation

Burgers' equation is a simplification of the Navier-Stokes equations, describing the speed of a fluid at a location in space and time (Bateman, 1915; Burgers, 1948). We consider Burgers' equation in one spatial dimension defined by the nonlinear PDE

$$u_t(s, t) = -u(s, t)u_x(s, t) + \nu u_{xx}(s, t),$$

where $u(s, t)$ is the speed of the fluid at location $s = (x)$ and time t and ν is the viscosity of the fluid. Data are generated using spectral differentiation and the *Tsit5* (Tsitouras, 2011) numerical solver from the Julia package *DifferentialEquations.jl* (Rackauckas and Nie, 2017) with initial condition $u(s, 0) = exp\{-(s + 2)^2\}$ and $\nu = 0.1$. The simulated data consist of 256 spatial locations across 101 time points where $D_s = [-8, 8]$ and $D_t = [0, 10]$ (Figure 2). We consider four cases – no noise, 2% noise, 5% noise, and 2% noise with 5% of data missing at random.
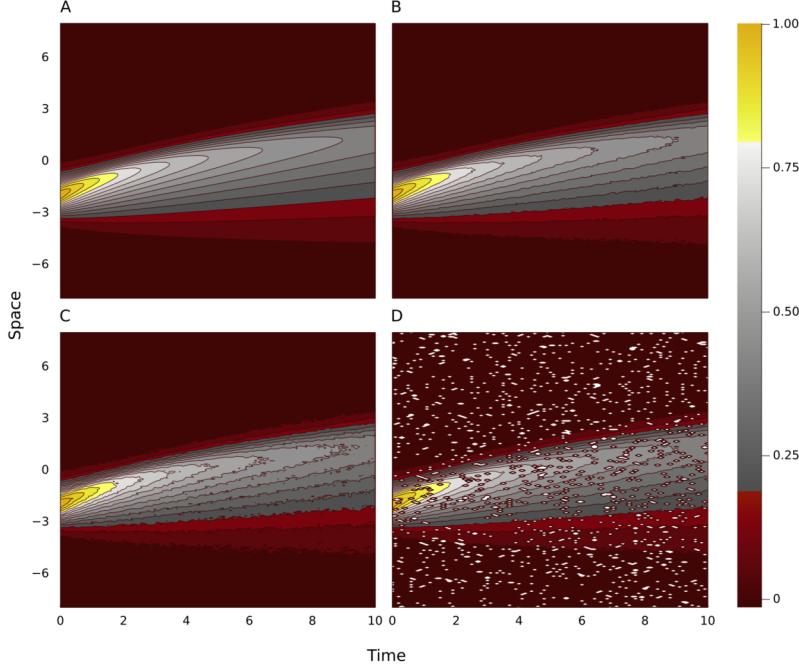
Figure 2: Data simulated from Burgers' equation with (A) no added noise, (B) 2% added noise, (C) 5% added noise, and (D) 5% of data missing at random and 2% noise.

For all four cases we specify the model with $P = 50$, $Q = 20$, $|\mathcal{Z}| = 100$, $\kappa = 10^{-4}$ and define the feature library as

$$[u, u^2, u^3, u_x, uu_x, u^2u_x, u^3u_x, u_{xx}, uu_{xx}, u^2u_{xx}, u^3u_{xx}, u_{xxx}, uu_{xxx}, u^2u_{xxx}, u^3u_{xxx}].$$

After obtaining posterior samples, we keep only terms with greater than a 50% inclusion probability to be included in the identified equation. The recovered equations and 95% highest posterior density (HPD) interval without and with noise for the included terms are shown in Table 2. In all four scenarios, the true components of the dynamic system are correctly identified. In addition, the probability of including extraneous terms from the feature library is low in each scenario (see Supplementary material Table 2), giving confidence to our identified equation.

The credible intervals of all parameters cover the true value with the exception of u_{xx} in the cases with 5% noise and 2% noise with 5% missing data. In addition, no extraneous terms are identified in any scenario. The probability of including another term (u_{xxx} for all four cases) is low, giving us relative certainty that the identified equation is indeed correct. Clearly, the methodology eventually will fail when measurement error is too large or there is too much missing data. For example, we found when the noise is greater than 8% or more than 10% of data are missing, we no longer recover the true model.

Noise	Missing Data	Statistic	Discovered Equation
0%	0%	Mean	$u_t = -0.994uu_x + 0.098u_{xx}$
		Lower HPD	$u_t = -1.022uu_x + 0.092u_{xx}$
		Upper HPD	$u_t = -0.964uu_x + 0.103u_{xx}$
2%	0%	Mean	$u_t = -0.990uu_x + 0.096u_{xx}$
		Lower HPD	$u_t = -1.033uu_x + 0.086u_{xx}$
		Upper HPD	$u_t = -0.954uu_x + 0.103u_{xx}$
5%	0%	Mean	$u_t = -0.981uu_x + 0.094u_{xx}$
		Lower HPD	$u_t = -1.022uu_x + 0.087u_{xx}$
		Upper HPD	$u_t = -0.951uu_x + 0.099u_{xx}$
2%	5%	Mean	$u_t = -0.957uu_x + 0.087u_{xx}$
		Lower HPD	$u_t = -1.003uu_x + 0.078u_{xx}$
		Upper HPD	$u_t = -0.931uu_x + 0.095u_{xx}$

Table 2: Discovered Burgers' equation (mean) and lower and upper HPD intervals with varying amounts of noise and missingness. The true Burgers' equation is $u_t = -uu_x + 0.1u_{xx}$.

5.2 Reaction-Diffusion Equation

The reaction-diffusion equation can be used to model the change in concentration or density of substances over time. We consider the 2D reaction-diffusion parameterized by the PDE

$$\mathbf{u}_t(\mathbf{s}, t) = \mathbf{W}\nabla^2\mathbf{u}(\mathbf{s}, t) + \mathbf{c}(\mathbf{u}(\mathbf{s}, t)),$$

where $\mathbf{u}(\mathbf{s}, t) = [b(\mathbf{s}, t), d(\mathbf{s}, t)]'$ may represent the concentration or density of two processes, $\mathbf{W}$ is a diagonal matrix of the diffusion coefficient for each process, and $\mathbf{c}(\cdot)$ is the (non)linear reaction function. The reaction-diffusion equation can be used to model the interaction between a predator and prey population (Hastings, 1996; Liu et al., 2019). To represent the interaction between prey and predator populations, we let $\mathbf{u}(\mathbf{s}, t) = [b(\mathbf{s}, t), d(\mathbf{s}, t)]'$ where $b(\mathbf{s}, t)$ and $d(\mathbf{s}, t)$ are the densities of the prey and predator populations, respectively. We define $\mathbf{c}(\cdot)$ to be the classic Lotka-Volterra model with a carrying capacity for the prey. Specifically,

$$\mathbf{c}(\cdot) = \begin{bmatrix} c_b(b(\mathbf{s}, t), d(\mathbf{s}, t)) \\ c_d(b(\mathbf{s}, t), d(\mathbf{s}, t)) \end{bmatrix} = \begin{bmatrix} \gamma_0 b(\mathbf{s}, t) - \frac{\gamma_0}{\gamma_1}b^2(\mathbf{s}, t) - \beta b(\mathbf{s}, t)d(\mathbf{s}, t) \\ \mu b(\mathbf{s}, t)d(\mathbf{s}, t) - \eta d(\mathbf{s}, t) \end{bmatrix},$$

where γ_0 is the prey growth rate, γ_1 is the prey carrying capacity, β predation rate, μ is the predator growth rate, and η is the predator death rate.

Suppressing the spatial and temporal indices, the predator-prey reaction-diffusion equation is

$$\begin{aligned} b_t &= W_b b_{xx} + W_b b_{yy} + \gamma_0 b - \frac{\gamma_0}{\gamma_1}b^2 - \beta bd, \\ d_t &= W_d d_{xx} + W_d d_{yy} + \mu bd - \eta d. \end{aligned} \tag{5.1}$$
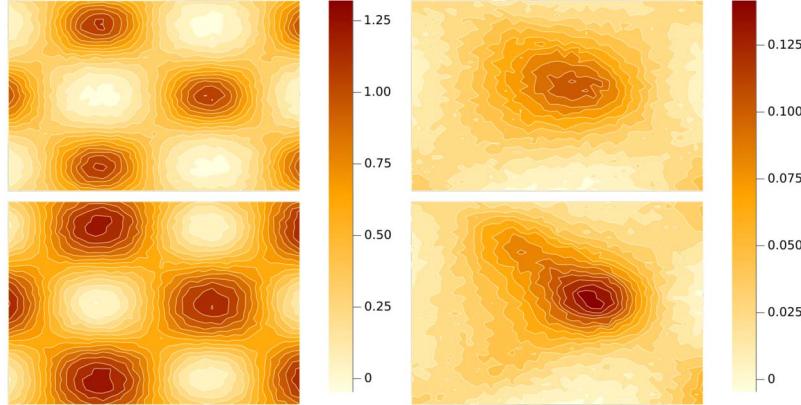
Figure 3: Data generated from the prey (left) and predator (right) reaction-diffusion system with 5% noise. Data are shown at time steps 11 (top, corresponding to $t = 1$) and 31 (bottom, corresponding to $t = 3$).

We simulate from (5.1) with $\gamma_0 = 0.4, \gamma_1 = 1.5, \beta = 0.5, \mu = 0.3, \eta = 0.1, W_b = 0.1, W_d = 0.1$ using a central finite difference scheme over the spatial domain $D_s = [-10, 10] \times [-10, 10]$ and the temporal domain $D_t = [0, 10]$ with a spatial and temporal resolution of $(0.5, 0.5)$ and 0.1, respectively. The prey and predator densities are initialized as

$$b(\mathbf{s}, 0) = \exp\{\cos(2\pi x/15)\sin(2\pi y/15)\},$$
$$d(\mathbf{s}, 0) = 0.1\exp\{\cos(2\pi y/30)\sin(2\pi x/30 - 5)\},$$

respectively. We again consider three scenarios – no noise, 2% noise, and 5% noise (Figure 3).

For all cases we specify our model with $P = 225, Q = 40, |\mathcal{Z}| = 100, \kappa = [10^{-4}, 10^{-6}]$ and define the feature library as

$$[b, b^2, b^3, d, d^2, d^3, bd, b^2d, bd^2, bb_x, bb_y, dd_x, dd_y, b_x, b_y, b_{xx}, b_{yy}, b_{xy}, d_x, d_y, d_{xx}, d_{yy}, d_{xy}].$$

Posterior estimates of terms with greater than 50% inclusion probability are shown in Tables 3, 4, and 5 for the case with no noise, 2% noise, and 5% noise, respectively. With no noise, we see all terms are correctly identified and the 95% HPD intervals all cover the truth. For the scenario with 2% noise, all terms are correctly identified and all coefficients except for b^2 in the prey equation cover the truth. The scenario with 5% noise correctly identifies all terms and only b^2 and b_{xx} for the prey equation and d_{xx} for the predator equation have 95% credible intervals that do not cover the truth.

6   Barotropic Vorticity Equation

Here we show the ability of our model to discover governing dynamic equations using real world data. The 500-hPa level of the atmosphere is often known as the "level

Noise	Component	Statistic	Recovered Equation
0%	Prey	Mean	$b_t = 0.400b - 0.266b^2 - 0.500bd + 0.099b_{xx} + 0.100b_{yy}$
		Lower	$b_t = 0.399b - 0.267b^2 - 0.503bd + 0.098b_{xx} + 0.099b_{yy}$
		Upper	$b_t = 0.400b - 0.266b^2 - 0.497bd + 0.100b_{xx} + 0.101b_{yy}$
0%	Predator	Mean	$d_t = -0.100d + 0.300bd + 0.099d_{xx} + 0.099d_{yy}$
		Lower	$d_t = -0.100d + 0.299bd + 0.099d_{xx} + 0.099d_{yy}$
		Upper	$d_t = -0.100d + 0.300bd + 0.100d_{xx} + 0.100d_{yy}$

Table 3: Discovered predator-prey reaction-diffusion equation (mean) and lower and upper HPD intervals with no noise. The true equations are $b_t = 0.4b - 0.26b^2 - 0.5bd + 0.1b_{xx} + 0.1b_{yy}$ and $d_t = 0.3bd - 0.1d + 0.1d_{xx} + 0.1d_{yy}$.

Noise	Component	Statistic	Recovered Equation
2%	Prey	Mean	$b_t = 0.403b - 0.270b^2 - 0.492bd + 0.144b_{xx} + 0.145b_{yy}$
		Lower	$b_t = 0.400b - 0.274b^2 - 0.501bd + 0.095b_{xx} + 0.097b_{yy}$
		Upper	$b_t = 0.405b - 0.267b^2 - 0.487bd + 0.163b_{xx} + 0.164b_{yy}$
2%	Predator	Mean	$d_t = -0.098d + 0.297bd + 0.136d_{xx} + 0.138d_{yy}$
		Lower	$d_t = -0.100d + 0.296bd + 0.095d_{xx} + 0.099d_{yy}$
		Upper	$d_t = -0.098d + 0.300bd + 0.161d_{xx} + 0.155d_{yy}$

Table 4: Discovered predator-prey reaction-diffusion equation (mean) and lower and upper HPD intervals with 2% noise. The true equations are $b_t = 0.4b - 0.26b^2 - 0.5bd + 0.1b_{xx} + 0.1b_{yy}$ and $d_t = 0.3bd - 0.1d + 0.1d_{xx} + 0.1d_{yy}$.

Noise	Component	Statistic	Recovered Equation
5%	Prey	Mean	$b_t = 0.401b - 0.269b^2 - 0.493bd + 0.090b_{xx} + 0.099b_{yy}$
		Lower	$b_t = 0.400b - 0.270b^2 - 0.504bd + 0.085b_{xx} + 0.091b_{yy}$
		Upper	$b_t = 0.403b - 0.267b^2 - 0.487bd + 0.093b_{xx} + 0.102b_{yy}$
5%	Predator	Mean	$d_t = -0.101d + 0.300bd + 0.092d_{xx} + 0.106d_{yy}$
		Lower	$d_t = -0.103d + 0.297bd + 0.086d_{xx} + 0.097d_{yy}$
		Upper	$d_t = -0.099d + 0.302bd + 0.095d_{xx} + 0.151d_{yy}$

Table 5: Discovered predator-prey reaction-diffusion equation (mean) and lower and upper HPD intervals with 5% noise. The true equations are $b_t = 0.4b - 0.26b^2 - 0.5bd + 0.1b_{xx} + 0.1b_{yy}$ and $d_t = 0.3bd - 0.1d + 0.1d_{xx} + 0.1d_{yy}$.

of non-divergence" because in the absence of strong cyclogenesis, the flow is essentially horizontal and non-divergent at this mid-level of the atmosphere. Such flows can often be modeled quite effectively with "barotropic" dynamics (in a barotropic fluid, the density is constant along a constant pressure surface). Indeed, the first successful numerical weather forecasts were based on the advection of relative vorticity (rotation of the fluid in the horizontal dimension) at the 500-hPa level using the so-called barotropic vorticity equation (BVE; Charney et al., 1950). The BVE is given as

$$\xi_t(\mathbf{s}, t) = -\mathbf{v}(\mathbf{s}, t) \cdot \nabla(\xi(\mathbf{s}, t) + f(\phi(\mathbf{s}))),$$

where $\xi = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$ is the relative vorticity, $\mathbf{v}(\mathbf{s}, t) = (u(\mathbf{s}, t), v(\mathbf{s}, t))$ is the non-divergent horizontal wind vector with $u(\mathbf{s}, t)$ the wind component in the zonal (east-west) direction, $v(\mathbf{s}, t)$ the meridional (north-south) wind component, and $f = 2\Omega \sin(lat(\mathbf{s}))$ is

the Coriolis parameter with $\Omega = 7.292 \times 10^{-5}$ radians per second, the angular speed of rotation of the Earth (note – not to be confused with the dynamic discovery library as defined previously), and $lat(\mathbf{s})$ the latitude in radians at location $\mathbf{s}$. The relative vorticity and non-divergent wind components can each be written in terms of the stream function, ψ. In particular, the vorticity is given in terms of the Laplacian of the streamfunction, $\xi = \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2}$, and the wind components are given by $v = \frac{\partial \psi}{\partial x}$, and $u = -\frac{\partial \psi}{\partial y}$ (e.g., Holton and Hakim, 2012). We can substitute these into the BVE to get an alternate form (suppressing the spatial and temporal indices):

$$\nabla^2 \psi_t = \psi_y \psi_{xxx} + \psi_y \psi_{xyy} - \psi_x \psi_{xxy} - \psi_x \psi_{yyy} - \psi_x f_y, \qquad (6.1)$$

where $\nabla^2 \psi_t = \psi_{xxt} + \psi_{yyt}$. This form is very useful for dynamic discovery because we need only consider functions of ψ. The first four terms on the RHS correspond to the advection of relative vorticity and the last term on the RHS corresponds to the advection of planetary vorticity (that associated with planetary rotation). Furthermore, note that the streamfunction can be computed based on the observed geopotential height, Φ, where $\psi(\mathbf{s}, t) = \Phi^*(\mathbf{s}, t)/f(lat(\mathbf{s}))$, $\Phi^*(\mathbf{s}, t) = \Phi(\mathbf{s}, t) - \int_D \Phi(\mathbf{s}, t)$, and D is the observed domain.
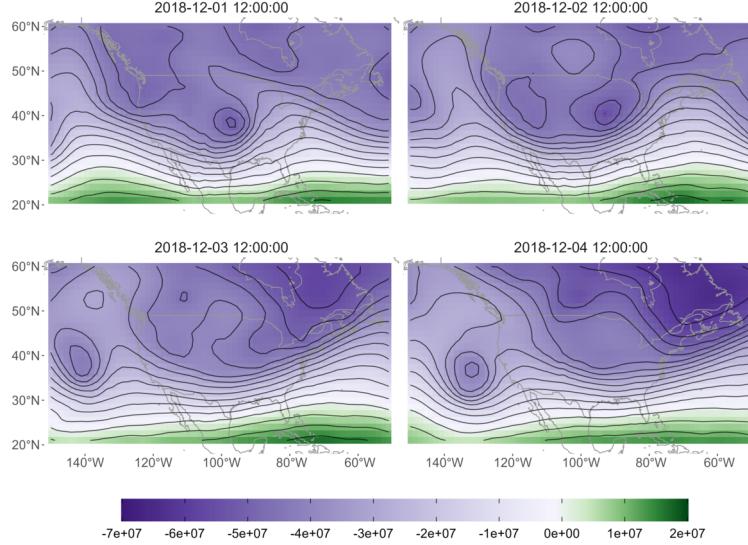
Figure 4: Streamfunction (m^2/s) data at 12:00 UTC on November 1, 2, 3, and 4, 2018. Purple and green correspond to lower value and upper values, respectively, with contour lines included for visual aid.

Here, we use hourly data generated using Copernicus Atmosphere Monitoring Service information (2022)[1] of relative geopotential height at 500-hPA. Data are collected hourly from December 1st to December 31, 2018 over the spatial domain $D_s =$

$[-150°W, -50°W] \times [20°N, 60°N]$ at a resolution of 1.5 degrees (see Figure 4 for example plots), resulting in $(67 \times 27) \times 744$ space-time locations. We compute the streamfunction from the geopotential height and use this as the observed data for the discovery. We also compute the derivative of the Coriolis parameter in the latitudinal direction, $f_y = 2\Omega \cos(lat(\mathbf{s}))$, and use it as a covariate in the model (i.e., $\omega(\mathbf{s}, t) = f_y(\mathbf{s})$ from (3.7)).

We specify our model with $P = 200$, $Q = 300$, $|\mathcal{Z}| = 200$, $\kappa = 10^{-4}$ and define the feature library as

$$[\psi, \psi_x, \psi_{xx}, \psi_y, \psi_{yy}, \psi_{xy}, \psi_x\psi_{xxx}, \psi_y\psi_{xxx}, \psi_x\psi_{yyy}, \psi_y\psi_{yyy},$$
$$\psi_x\psi_{xxy}, \psi_y\psi_{xxy}, \psi_x\psi_{xyy}, \psi_y\psi_{xyy}, \psi_x f_y, \psi_y f_y].$$

Posterior estimates for the recovered equation are shown in Table 6, where only terms with a posterior inclusion probability greater than 50% were kept. While we do not know the true equation in this case (because the barotropic vorticity equation is only an approximation of the dynamics in the atmosphere), we see the discovered equation closely resembles the hypothesized BVE given in (6.1).

Although the sign for each discovered term aligns with the sign in the BVE, we note that the coefficient values are different from the hypothesized equation and the planetary vorticity term ($\psi_x f_y$) is not significant. There are several reasons why this is likely the case and not surprising. First, it is possible that this particular time period is not barotropic (e.g., baroclinic), which would require library terms our current framework cannot accommodate (e.g., flow in the vertical direction, density, temperature). Such transient changes from barotropic to baroclinic flow regimes are common in the upper-level atmospheric flow. The lack of significance of the planetary vorticity term is also interesting. Recall that the meridional velocity is related to the gradient of streamfunction in the meridional direction, so this term can be alternatively written $\psi_x f_y = v f_y$. Thus, if the flow is primarily zonal (east-west), the north-south flow component (v) is quite small and this term may not be important for a particular time period. Indeed, visual examination of the data suggests that the flow is dominated by zonal flow and this is further confirmed seeing that the magnitude of the planetary vorticity term in the BVE is an order of magnitude smaller compared to the relative vorticity terms. Yet, the planetary vorticity term is still important to the advection of relative vorticity because of the Coriolis effect on the flow, resulting in the term being identified as important (if not significant). We also note that the planetary vorticity term in the BVE is important for supporting so-called Rossby waves (slowly moving westward propagating synoptic-scale waves) in the upper level flow. Yet, the time period and spatial extent of the data considered here are likely not sufficient to capture the full spatial and temporal

Statistic	Discovered Equation
Mean	$\nabla^2\psi_t = 0.289\psi_y\psi_{xxx} + 0.277\psi_y\psi_{xyy} - 0.280\psi_x\psi_{xxy} - 0.185\psi_x\psi_{yyy} - 6.354\psi_x f_y$
Lower HPD	$\nabla^2\psi_t = 0.235\psi_y\psi_{xxx} + 0.267\psi_y\psi_{xyy} - 0.343\psi_x\psi_{xxy} - 0.215\psi_x\psi_{yyy} - 7.570\psi_x f_y$
Upper HPD	$\nabla^2\psi_t = 0.317\psi_y\psi_{xxx} + 0.286\psi_y\psi_{xyy} - 0.223\psi_x\psi_{xxy} - 0.160\psi_x\psi_{yyy} + 1.491\psi_x f_y$

Table 6: Discovered equation for the BVE (mean) and lower and upper HPD intervals where the theoretical BVE is $\nabla^2\psi_t = \psi_y\psi_{xxx} + \psi_y\psi_{xyy} - \psi_x\psi_{xxy} - \psi_x\psi_{yyy} - \psi_x f_y$.

scale of Rossby waves (e.g., Holton and Hakim, 2012). Thus, it is somewhat remarkable that the dynamic discovery methodology was still able to infer realistic properties of the system of interest that generally align with the theoretical system. To the best of our knowledge, this is the first time that data-driven discovery methods have been applied to real-world atmospheric data and have identified physically plausible features.

7   Conclusion

We have proposed a data-driven approach for learning complex nonlinear spatio-temporal dynamic equations that is robust to measurement error and missing data. Our approach uses a Bayesian hierarchical model where the dynamic equation is embedded in the latent process enabling the discovery of dynamic equations within the statistical framework. Additionally, the model provides probabilistic estimates of inclusion for each component of the feature library and estimates of uncertainty for the recovered parameters, giving a deeper understanding to the dynamic system. This all stems from the expansion of the dynamic process in terms of basis functions, bypassing the need for numerical differentiation and enabling the estimate of the derivatives within a probabilistic framework.

Improvements to the current framework should focus on the specification of the feature library and the selection prior. A feature library that is uninhibited by the users choice (i.e., the model could generate library terms) would remove user bias. This is akin to what has been proposed with symbolic regression. Also, a different choice in selection prior for the coefficients, perhaps one that also penalizes model complexity, has the potential to improve selection performance. While there are a variety of selection priors in the literature, a prior directed towards this problem will provide noticeable improvement on the identification of the system.

From the application perspective, the promising and intriguing results from the BVE example suggest further investigation. In particular, it would be informative to consider larger spatial domains (hemispheric) and temporal time periods (many months) to determine if the presence of Rossby waves are sufficient to lead to a significant planetary vorticity term. This also suggests experiments with numerical simulation models that can easily be controlled to exhibit barotropic or baroclinic flow. Such models will allow us to investigate the effects of the flow regime on the ability to detect realistic dynamics.

As a final point, our approach could be used to aid in a decision-based framework. Dynamic equations provide a bridge between Granger and Pearl causality, where the dynamic equation dually represents the graphical structure from the Pearl pedagogy and the dynamic component from the Granger pedagogy. By identifying terms of the dynamic equation, or put another way, the variables forcing the response (e.g., forcing of green house gasses on temperature), then one can infer the cause and effect relationship of the system. Additionally, the system can be estimated multiple times where "important" drivers of the response are omitted from the feature library to identify which other potential drivers of the system. This could be used in a decision framework, where, now knowing the underlying dynamics driving the system, the effects of various inputs can be assessed by their impact on the response.

Supplementary Material

Supporting Material for "A Bayesian Approach for Spatio-Temporal Data-Driven Dynamic Equation Discovery" (DOI: 10.1214/23-BA1406SUPP; .pdf). The Supplementary Material contains a discussion of how to select the subsample size, full conditional distributions and sampling algorithm, example of the method for the heat equation, and proofs of Propositions 1 and 2.

References

Bateman, H. (1915). "Some recent researches on the motion of fluids." *Monthly Weather Review*, 43(4): 163–170. MR1524531. doi: https://doi.org/10.2307/2302468. 17

Berliner, L. M. (1996). "Hierarchical Bayesian time series models." In *Maximum Entropy and Bayesian Methods*, 15–22. Dordrecht: Springer Netherlands. MR1446713. 3

Berliner, L. M. (2003). "Physical-statistical modeling in geophysics." *Journal of Geophysical Research: Atmospheres*, 108(D24). 3

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). "Julia: A fresh approach to numerical computing." *SIAM Review*, 59(1): 65–98. MR3605826. doi: https://doi.org/10.1137/141000671. 14

Bolker, B. and Grenfell, B. (1995). "Space, persistence and dynamics of measles epidemics." *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 348(1325): 309–320. 1

Bongard, J. and Lipson, H. (2007). "Automated reverse engineering of nonlinear dynamical systems." *Proceedings of the National Academy of Sciences*, 104(24): 9943–9948. 2

Boninsegna, L., Nüske, F., and Clementi, C. (2018). "Sparse learning of stochastic dynamical equations." *The Journal of Chemical Physics*, 148(24): 241723. 2

Both, G.-J., Choudhury, S., Sens, P., and Kusters, R. (2021). "DeepMoD: Deep learning for model discovery in noisy data." *Journal of Computational Physics*, 428(1): 109985. MR4199381. doi: https://doi.org/10.1016/j.jcp.2020.109985. 2

Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2016). "Discovering governing equations from data by sparse identification of nonlinear dynamical systems." *Proceedings of the National Academy of Sciences*, 113(15): 3932–3937. MR3494081. doi: https://doi.org/10.1073/pnas.1517384113. 2

Burgers, J. (1948). "A mathematical model illustrating the theory of turbulence." In *Advances in Applied Mechanics*, volume 1, 171–199. Elsevier. MR0027195. 17

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010). "The horseshoe estimator for sparse signals." *Biometrika*, 97(2): 465–480. MR2650751. doi: https://doi.org/10.1093/biomet/asq017. 11

Champion, K., Zheng, P., Aravkin, A. Y., Brunton, S. L., and Kutz, J. N. (2020).

"A unified sparse optimization framework to learn parsimonious physics-informed models from data." *IEEE Access*, 8: 169259–169271.   2

Charney, J. G., FjÖrtoft, R., and Neumann, J. V. (1950). "Numerical integration of the barotropic vorticity equation." *Tellus*, 2(4): 237–254. MR0042799. doi: `https://doi.org/10.3402/tellusa.v2i4.8607`.   1, 21

Cressie, N. A. C. and Wikle, C. K. (2011). *Statistics for spatio-temporal data*. John Wiley & Sons. MR2848400.   3, 10, 13

de Silva, B., Champion, K., Quade, M., Loiseau, J.-C., Kutz, J., and Brunton, S. (2020). "PySINDy: A Python package for the sparse identification of nonlinear dynamical systems from data." *Journal of Open Source Software*, 5(49): 2104.   2

Fasel, U., Kutz, J. N., Brunton, B. W., and Brunton, S. L. (2021). "Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control." *ArXiv*, 1–18. MR4416372. doi: `https://doi.org/10.1098/rspa.2021.0904`.   2

Gelman, A. (2006). "Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper)." *Bayesian Analysis*, 1(3): 515–534. MR2221284. doi: `https://doi.org/10.1214/06-BA117A`.   13

George, E. I., McCulloch, R. E., George, E. I., and McCulloch, R. E. (1993). "Variable selection via Gibbs sampling." *Journal of the American Statistical Association*, 88(423): 881–889.   11

Hastings, A. (1996). "Models of spatial spread: Is the theory complete?" *Ecology*, 77(6): 1675–1679.   1, 7, 19

Higham, N. J., Dennis, M. R., Glendinning, P., Martin, P. A., Santosa, F., and Tanner, J. (2016). *The Princeton companion to applied mathematics*. Princeton University Press. MR3380576. doi: `https://doi.org/10.1515/9781400874477`.   1, 9

Hirsh, S. M., Barajas-Solano, D. A., and Kutz, J. N. (2021). "Sparsifying priors for Bayesian uncertainty quantification in model discovery." *ArXiv*, 1–22.   2

Holton, J. R. and Hakim, G. J. (2012). *An introduction to dynamic meteorology*, 5th edition. Academic Press.   1, 22, 24

Hooten, M. B. and Wikle, C. K. (2008). "A hierarchical Bayesian non-linear spatio-temporal model for the spread of invasive species with application to the Eurasian Collared-Dove." *Environmental and Ecological Statistics*, 15(1): 59–70. MR2412679. doi: `https://doi.org/10.1007/s10651-007-0040-1`.   3

Huang, A. and Wand, M. P. (2013). "Simple marginally noninformative prior distributions for covariance matrices." *Bayesian Analysis*, 8(2): 439–452. MR3066948. doi: `https://doi.org/10.1214/13-BA815`.   13

Kolda, T. (2006). "Multilinear operators for higher-order decompositions." Technical Report April, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States).   5

Kolda, T. G. and Bader, B. W. (2009). "Tensor decompositions and applications." *SIAM Review*, 51(3): 455–500. MR2535056. doi: https://doi.org/10.1137/07070111X. 4

Kühnert, D., Stadler, T., Vaughan, T. G., and Drummond, A. J. (2014). "Simultaneous reconstruction of evolutionary history and epidemiological dynamics from viral sequences with the birth–death SIR model." *Journal of The Royal Society Interface*, 11(94): 20131106. 1

Kuhnert, P. M. (2017). "Physical-statistical modeling." In *Wiley StatsRef: Statistics Reference Online*, 1–5. Wiley. MR0144404. 3

Li, Q. and Lin, N. (2010). "The Bayesian elastic net." *Bayesian Analysis*, 5(1): 151–170. MR2596439. doi: https://doi.org/10.1214/10-BA506. 13

Liu, H., Ye, Y., Wei, Y., Ma, W., Ma, M., and Zhang, K. (2019). "Pattern formation in a reaction-diffusion predator-prey model with weak allee effect and delay." *Complexity*, 2019(1): 1–14. MR3994110. doi: https://doi.org/10.1155/2019/7296461. 1, 19

Long, Z., Lu, Y., and Dong, B. (2019). "PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network." *Journal of Computational Physics*, 399: 108925. MR4013148. doi: https://doi.org/10.1016/j.jcp.2019.108925. 2

Long, Z., Lu, Y., Ma, X., and Dong, B. (2017). "PDE-Net: Learning PDEs from data." In *35th International Conference on Machine Learning, ICML 2018*, 7: 5067–5078. 2

Malsiner-Walli, G. and Wagner, H. (2016). "Comparing spike and slab priors for Bayesian variable selection." *Austrian Journal of Statistics*, 40(4). 11

Mandt, S., Hoffman, M., and Blei, D. (2016). "A variational analysis of stochastic gradient algorithms." In: *Proceedings of The 33rd International Conference on Machine Learning*, 48: 354–363. 13, 14

Mangal, T. D., Paterson, S., and Fenton, A. (2008). "Predicting the impact of long-term temperature changes on the epidemiology and control of Schistosomiasis: A mechanistic model." *PLoS ONE*, 3(1): e1438. MR2575022. doi: https://doi.org/10.1371/journal.pcbi.1000536. 1

Milliff, R. F., Bonazzi, A., Wikle, C. K., Pinardi, N., and Berliner, L. M. (2011). "Ocean ensemble forecasting. Part I: Ensemble Mediterranean winds from a Bayesian hierarchical model." *Quarterly Journal of the Royal Meteorological Society*, 137(657): 858–878. 3

Mitchell, T. J. and Beauchamp, J. J. (1988). "Bayesian variable selection in linear regression." *Journal of the American Statistical Association*, 83(404): 1023. MR0997578. 11

Niven, R., Mohammad-Djafari, A., Cordier, L., Abel, M., and Quade, M. (2020). "Bayesian identification of dynamical systems." *Proceedings*, 33(1): 33. 2

North, J. S., Wikle, C. K., and Schliep, E. M. (2022). "A Bayesian approach for data-driven dynamic equation discovery." *Journal of Agricultural, Biological, and Environ-*

mental Statistics, 1(1): 1–28. MR4498007. doi: https://doi.org/10.1007/s13253-022-00514-1. 3, 15

North, J. S., Wikle, C. K., and Schliep, E. M. (2023). "Supplementary Material for "A Bayesian approach for spatio-temporal data-driven dynamic equation discovery"." *Bayesian Analysis*. doi: https://doi.org/10.1214/23-BA1406SUPP. 9

Park, T. and Casella, G. (2008). "The Bayesian lasso." *Journal of the American Statistical Association*, 103(482): 681–686. MR2524001. doi: https://doi.org/10.1198/016214508000000337. 11

Rackauckas, C. and Nie, Q. (2017). "DifferentialEquations.jl – A performant and feature-rich ecosystem for solving differential equations in Julia." *Journal of Open Research Software*, 5(1): 15. 17

Raissi, M. and Karniadakis, G. E. (2018). "Hidden physics models: Machine learning of nonlinear partial differential equations." *Journal of Computational Physics*, 357: 125–141. MR3759415. doi: https://doi.org/10.1016/j.jcp.2017.11.039. 2

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2017). "Physics informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations." Part I: 1–22. *arXiv preprint arXiv:1711.10561*, 2

Raissi, M., Yazdani, A., and Karniadakis, G. E. (2020). "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations." *Science*, 367(6481): 1026–1030. MR4265157. doi: https://doi.org/10.1126/science.aaw4741. 2

Ramsay, J. O. and Silverman, B. W. (2005). *Functional data analysis*. Springer Series in Statistics. New York, NY: Springer New York. MR2168993. 15

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat (2019). "Deep learning and process understanding for data-driven Earth system science." *Nature*, 566(7743): 195–204. 2

Revels, J., Lubin, M., and Papamarkou, T. (2016). "Forward-mode automatic differentiation in Julia." *ArXiv*. 14

Ročková, V. and George, E. I. (2014). "Negotiating multicollinearity with spike-and-slab priors." *METRON*, 72(2): 217–229. MR3233150. doi: https://doi.org/10.1007/s40300-014-0047-y. 11

Royle, J. A., Berliner, L. M., Wikle, C. K., and Milliff, R. (1999). "A Hierarchical spatial model for constructing wind fields from scatterometer data in the Labrador Sea." In *Case Studies in Bayesian Statistics.*, 367–382. Springer, New York, NY. 3

Rudy, S., Alla, A., Brunton, S. L., and Kutz, J. N. (2019a). "Data-driven identification of parametric partial differential equations." *SIAM Journal on Applied Dynamical Systems*, 18(2): 643–660. MR3936890. doi: https://doi.org/10.1137/18M1191944. 2

Rudy, S. H., Brunton, S. L., Proctor, J. L., and Kutz, J. N. (2017). "Data-driven discovery of partial differential equations." *Science Advances*, 3(4): e1602614. 2

Rudy, S. H., Nathan Kutz, J., and Brunton, S. L. (2019b). "Deep learning of dynamics and signal-noise decomposition with time-stepping constraints." *Journal of Computational Physics*, 396: 483–506. MR3989622. doi: https://doi.org/10.1016/j.jcp.2019.06.056. 2

Schaeffer, H. (2017). "Learning partial differential equations via data discovery and sparse optimization." *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197): 20160446. MR3615300. doi: https://doi.org/10.1098/rspa.2016.0446. 2

Schmidt, M. and Lipson, H. (2009). "Distilling free-form natural laws from experimental data." *Science*, 324(5923): 81–85. 2

Sun, Y., Zhang, L., and Schaeffer, H. (2019). "NeuPDE: Neural network based ordinary and partial differential equations for modeling time-dependent data." 107(2016): 352–372. *arXiv preprint arXiv:1908.03190*, 2

Tibshirani, R. (1996). "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1): 267–288. MR1379242. 2

Tsitouras, C. (2011). "Runge–Kutta pairs of order 5(4) satisfying only the first column simplifying assumption." *Computers & Mathematics with Applications*, 62(2): 770–775. MR2817913. doi: https://doi.org/10.1016/j.camwa.2011.06.002. 17

Tucker, L. R. (1966). "Some mathematical notes on three-mode factor analysis." *Psychometrika*, 31(3): 279–311. MR0205395. doi: https://doi.org/10.1007/BF02289464. 5

Wang, J. L., Chiou, J. M., and Müller, H. G. (2016). "Functional data analysis." *Annual Review of Statistics and Its Application*, 3: 257–295. 9

White, F. M. and Majdalani, J. (2006). *Viscous fluid flow*, 3rd edition. McGraw-Hill, New York. 1

Wikle, C. K. (2003). "Hierarchical Bayesian models for predicting the spread of ecological processes." *Ecology*, 84(6): 1382–1394. 3

Wikle, C. K., Milliff, R. F., Nychka, D., and Berliner, L. M. (2001). "Spatiotemporal hierarchical Bayesian modeling: Tropical ocean surface winds." *Journal of the American Statistical Association*, 96(454): 382–397. MR1939342. doi: https://doi.org/10.1198/016214501753168109. 3

Wu, K. and Xiu, D. (2020). "Data-driven deep learning of partial differential equations in modal space." *Journal of Computational Physics*, 408: 109307. MR4065433. doi: https://doi.org/10.1016/j.jcp.2020.109307. 2

Xu, H., Chang, H., and Zhang, D. (2019). "DL-PDE: Deep-learning based data-driven discovery of partial differential equations from discrete and noisy data." *Communications in Computational Physics*, 29(3): 698–728. MR4203108. doi: https://doi.org/10.4208/cicp.oa-2020-0142. 2

Xu, H., Chang, H., and Zhang, D. (2020). "DLGA-PDE: Discovery of PDEs with incomplete candidate library via combination of deep learning and genetic algorithm."

Journal of Computational Physics, 418: 109584. MR4127289. doi: https://doi.org/10.1016/j.jcp.2020.109584. 2

Xu, H., Zhang, D., and Zeng, J. (2021). "Deep-learning of parametric partial differential equations from sparse and noisy data." *Physics of Fluids*, 33(3): 037132. 2

Yang, H.-C., Hu, G., and Chen, M.-H. (2019). "Bayesian variable selection for pareto regression models with latent multivariate log gamma process with applications to earthquake magnitudes." *Geosciences*, 9(4): 169. 2

Yang, Y., Aziz Bhouri, M., and Perdikaris, P. (2020). "Bayesian differential programming for robust systems identification under uncertainty." *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2243): 20200290. MR4187941. doi: https://doi.org/10.1098/rspa.2020.0290. 3

Zellner, A. (1986). "On assessing prior distributions and Bayesian regression analysis with g-prior distributions." *Bayesian Inference and Decision Techniques*, 233–243. MR0881437. 11

Zhang, S. and Lin, G. (2018). "Robust data-driven discovery of governing physical laws with error bars." *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2217): 20180305. MR3875202. doi: https://doi.org/10.1098/rspa.2018.0305. 2

Zheng, P., Askham, T., Brunton, S. L., Kutz, J. N., and Aravkin, A. Y. (2019). "A unified framework for sparse relaxed regularized regression: SR3." *IEEE Access*, 7: 1404–1423. 2