Discrepancy Modeling Framework: Learning missing physics, modeling systematic residuals, and disambiguating between deterministic and random effects

Megan R. Ebers[1], Katherine M. Steele[1], and J. Nathan Kutz[2]

[1]Department of Mechanical Engineering, University of Washington
[2]Department of Applied Mathematics, University of Washington

March 11, 2022

Abstract

Physics-based and first-principles models pervade the engineering and physical sciences, allowing for the ability to model the dynamics of complex systems with a prescribed accuracy. The approximations used in deriving governing equations often result in discrepancies between the model and sensor-based measurements of the system, revealing the approximate nature of the equations and/or the signal-to-noise ratio of the sensor itself. In modern dynamical systems, such discrepancies between model and measurement can lead to poor quantification, often undermining the ability to produce accurate and precise control algorithms. We introduce a discrepancy modeling framework to resolve deterministic model-measurement mismatch with two distinct approaches: (i) by learning a model for the evolution of systematic state-space residual, and (ii) by discovering a model for the missing deterministic physics. Regardless of approach, a common suite of data-driven model discovery methods can be used. Specifically, we use four fundamentally different methods to demonstrate the mathematical implementations of discrepancy modeling: (i) the *sparse identification of nonlinear dynamics* (SINDy), (ii) *dynamic mode decomposition* (DMD), (iii) *Gaussian process regression* (GPR), and (iv) *neural networks* (NN). The choice of method depends on one's *intent* for discrepancy modeling, as well as *quantity* and *quality* of the sensor measurements. We demonstrate the utility and suitability for both discrepancy modeling approaches using the suite of data-driven modeling methods on three dynamical systems under varying signal-to-noise ratios. We compare reconstruction and forecasting accuracies and provide detailed comparatives, allowing one to select the appropriate approach and method in practice.

1 Introduction

The traditional modeling of physics and engineering systems relies on the development of governing equations that characterize the underlying nonlinear, dynamical processes. Such governing equations are typically derived through asymptotic reductions, enforcing physical constraints or conservation laws, and/or positing empirical relations between variables [1]. Simulation of the governing equations then allows for prediction, control, and characterization of the complex system. However, it is well known that governing equations are often idealized and only approximate, either achieved through dominant balance physics arguments or neglecting higher-order effects [2–5]. In many emerging fields, the idealized models currently used are simply inadequate for modeling applications where precision is necessary, such as in robotics, biomechanics, precision manufacturing, and automated systems. The time evolution of complex nonlinear systems is often highly sensitive to small errors in system dynamics. This limits the utility of simulation, such as for control or inference. Generally, there are two kinds of errors that occur in modeling physical systems: missing physics and noise. In practice, both errors exist and are difficult to disambiguate. Our framework provides approaches which help disambiguate between the dominant error forms, thereby learning missing physics and/or characterizing the residual between models and measurements.

With the advancement of modern sensor technologies, for both increased quality and quantity, there is opportunity to improve the characterization of system dynamics through modern data-driven methods to refine and augment the known, governing first-principles. Indeed, a better understanding of the underlying physical processes can be achieved by inspecting the residual between first-principles theory and sensor measurements of dynamical systems. The residual may contain deterministic effects, or *discrepancies*, and models of discrepancy dynamics can be learned using data-driven model discovery. A number of machine learning techniques have been developed to learn model error or discrepancies using hybrid data assimilation techniques. Thus they learn additive correction models for the missing physics [6–9] for a diversity of applications. Levine and Stuart [10] have also recently proposed leveraging machine learning methods, specifically recurrent neural networks, for modeling the discrepancies between a given model and measurements of the system. By augmenting known first-principles with a discrepancy model, an improved deterministic model can be learned. Although we have always improved our models through systematic approaches, we build on recent work [10, 11] and outline here two principled, data-driven approaches by which discrepancy modeling is automated. Moreover, we examine the relationship between deterministic and random effects within the residual; experimental noise in sensor measurements dictate limitations in learning a discrepancy model.

Discrepancy modeling has deep historical context, especially since early models of any system are typically coarse approximations of the physics. Indeed, throughout the 1960s, before the advancement of scientific computing, asymptotic and perturbation methods [2, 3] that systematically introduced discrepancies were leading methods in the development of fluid dynamics. From the Prandtl number to the Reynolds number, various approximations led to different dominant balance physics approximations [4, 5, 12]. Thus by construction, such models allowed for improved understanding at the expense of detailed models. Computation has allowed us to move beyond such reductions, yet model-measurement mismatch continues to prevent accurate interpretation of system structure and/or prediction of time evolution. Indeed, in almost every application area, mismatch exists between experiment and theory which is not due to noise. For instance, improvements in tracking planetary motion in the late 1800s and early 1900s allowed for the characterization of a discrepancy between Newton's gravitation laws and the observed physics, eventually leading to the development of general relativity by Einstein [13]. More recently, identifying missing deterministic effects (provided they are not obscured by noise) has allowed for the discovery of missing physics that are challenging to model with first principles, such as fluid drag forces of falling objects [14] or bearing chatter during double pendulum control [11]. Our goal is to leverage improved sensor observations; we automate the process of building better models by identifying a discrepancy model.

The ideas in our discrepancy modeling framework are well established concepts in statistics and optimization [15–19]. More specifically, in statistical regression analysis, measures of deviation are referred to as either *error* or *residual*. An *error* is the difference between the observed values and the true values. It is important to note that true values are unobservable and thus error values are inaccessible. A *residual* is the difference between the observed values and the estimated

Variable	Description
$\mathbf{x}$	approximate state space
$\tilde{\mathbf{x}}$	augmented state space
$\mathbf{x}_0$	true state space (inaccessible)
$\mathbf{y}_k$	measurements
$f(\cdot)$	approximate dynamics
$\tilde{F}(\cdot)$	augmented dynamics
$F(\cdot)$	true dynamics
$G(\cdot)$	discrepancy dynamics
$\mathbf{E}(t) = \mathbf{y}_k - \mathbf{x}$	state space residual
$\tilde{\mathbf{E}}(t) = \mathbf{y}_k - \tilde{\mathbf{x}}$	augmented state space residual
$\mathbf{E}_f(t) = \dot{\mathbf{y}}_k - f(\mathbf{y}_k)$	dynamical error
$\tilde{\mathbf{E}}_f(t) = \dot{\mathbf{y}}_k - \tilde{F}(\mathbf{y}_k)$	augmented dynamical error

Table 1: Variable definitions. The goal of discrepancy modeling is to compute $\tilde{\mathbf{x}}$ and by discovering $G(\cdot)$. There are two ways to calculate a discrepancy: learning the dynamical error, $\mathbf{E}_f(t)$, or modeling the systematic residual, $\mathbf{E}(t)$.

values. In practice, observed values are used as a proxy for the true values; therefore, residuals may contain both random and deterministic signals. Regression analysis seeks to evaluate how well a statistical model fits a data set; if the residual contains a bias, it suggests the model can be improved by capturing deterministic values within the residual. We posit that discrepancy modeling is to dynamical systems modeling what regression analysis is to statistical modeling. In the context of dynamical systems, traditional approaches to discrepancy modeling include Kalman filtering and data-assimilation; however, they assume the mismatch between model and measurement are given by normally-distributed variables, *i.e.*, random processes [20, 21]. However, in many systems the residual error may reveal deterministic structure that dictate the observed time evolution. As already noted, Levine and Stuart [10] recently have advanced the state-of-the-art by leveraging machine learning methods (neural networks) for learning missing physics. More broadly, they provide a rigorous analysis for learning such models from data in dynamical systems settings. In this work, we build on this theme by considering a broader class of models, including those constructed by *sparse identification of dynamical systems* (SINDy), *dynamic mode decomposition* (DMD), and *Gaussian process regression* (GPR). We also advocate learning a model for the systematic residual directly from data measurements. Again, we use SINDy, DMD, and GPR along with neural networks (NN) to learn models of the residual directly and correct our state-space solutions. This hybrid framework brings together domain knowledge from first principles and data-driven model discovery to provide a more comprehensive modeling space [22].

From a modeling perspective (see definitions in Table 1), it is assumed that an approximate model of the physics is available, *i.e.*, the Platonic model:

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, t) \tag{1}$$

where the governing dynamics $f(\mathbf{x}, t)$ are known and derived from first principles, asymptotic reductions, enforcing physical constraints or conservation laws, or positing empirical relations between variables. However, in truth, the true dynamics $F(\cdot)$, *which we do not have access to*, are given by

$$\frac{d\mathbf{x}_0}{dt} = F(\mathbf{x}_0, t) = f(\mathbf{x}_0, t) + g(\mathbf{x}_0, t) + \mathcal{N}_1(\mu_1, \sigma_1) \tag{2}$$

where $g(\mathbf{x}_0, t)$ is the deterministic physics that remains unmodeled due to some suitable approximation and/or lack of physics knowledge, while $\mathcal{N}_1(\mu_1, \sigma_1)$ is a noise process that drives stochastic variability in the model. Note that the missing dynamics comprising $g(\mathbf{x}_0, t)$ may be intentionally omitted (*e.g.*, through model reduction) or unintentionally omitted (*e.g.*, due to lack of first-principles knowledge of the system structure). The combination of $g(\mathbf{x}_0, t) + \mathcal{N}_1(\mu_1, \sigma_1)$ is the residual. Thus, these two terms would need to be disambiguated in order to learn a discrepancy model of the deterministic effect $g(\mathbf{x}_0, t)$.

Measurements of the system are made at discrete time points so that

$$\mathbf{y}_k = \mathbf{y}(t_k) = \mathbf{x}_0(t_k) + \mathcal{N}_2(\mu_2, \sigma_2) \tag{3}$$

where $\mathcal{N}_2(\mu_2, \sigma_2)$ is a noise process describing observation (or sensor) noise. The goal of discrepancy modeling is to improve (1). This can be done in two distinct ways. First, one can generate an improved dynamical model:

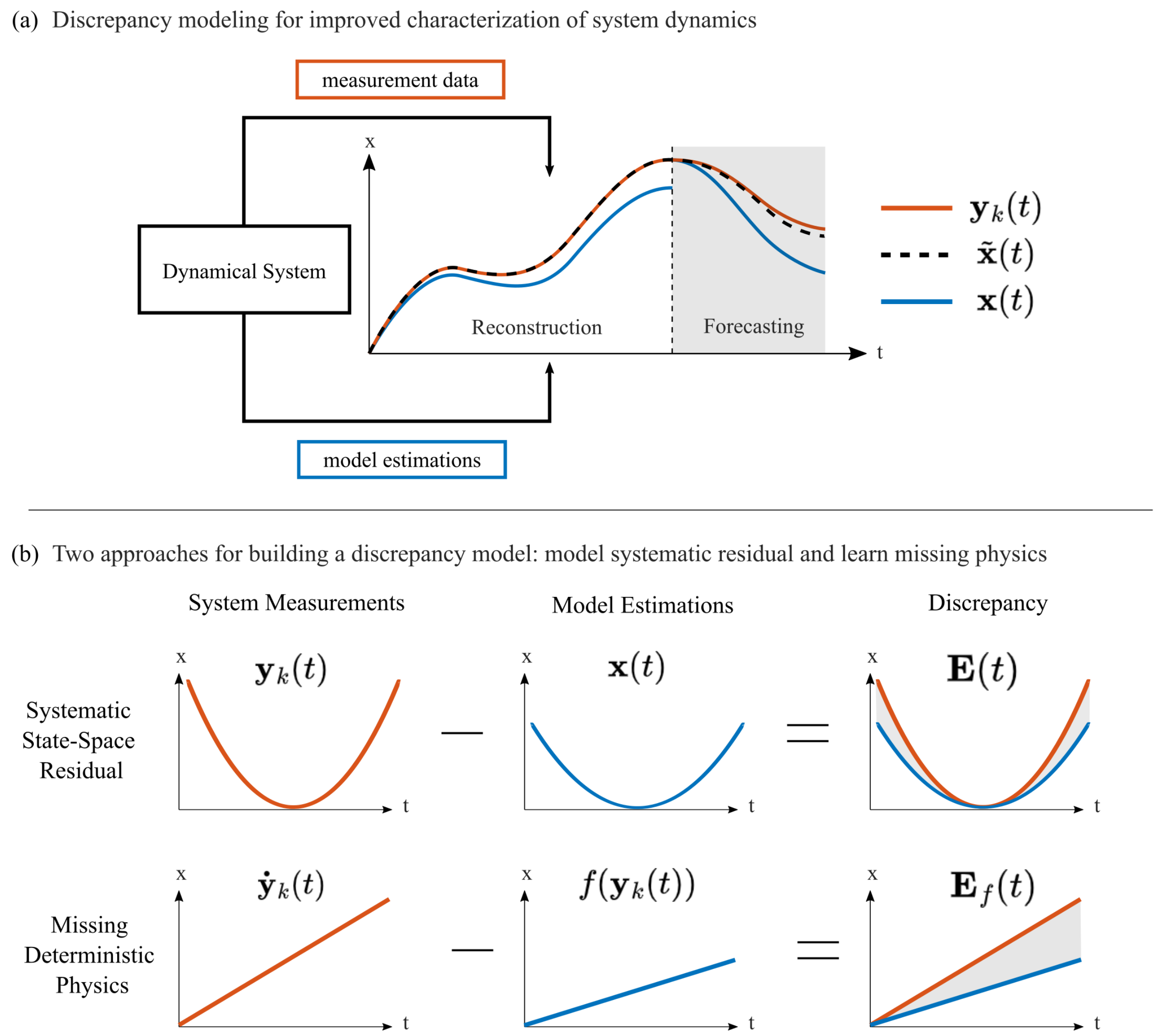
$$\frac{d\tilde{\mathbf{x}}}{dt} = \tilde{F}(\tilde{\mathbf{x}}, t) = f(\tilde{\mathbf{x}}, t) + G(\tilde{\mathbf{x}}, t) \tag{4}$$

such that $\mathbf{y}_k - \tilde{\mathbf{x}} < \mathbf{y}_k - \mathbf{x}$, *i.e.*, the augmented dynamics are less erroneous than the approximate dynamics. In a second method, given the state space residual $\mathbf{E}(t)$, one can decrease the state space residual by correcting the approximate dynamics:

$$\tilde{\mathbf{x}}(t) = \mathbf{x}(t) + G(\mathbf{x}, t). \tag{5}$$

Note that in both methods, $\tilde{\mathbf{x}}$ is the augmented and improved state space, while $G(\cdot)$ is the discrepancy model. In the former approach, missing physics are learned, while in the latter approach, a model for the residual is constructed. Equations (4) and (5) are explicitly the focus of this manuscript. Table 1 summarizes all our variable definitions whereas Fig. 1 illustrates the discrepancy modeling framework.

By focusing explicitly on the discrepancy between measured and modeled dynamics, our framework shifts the view of discrepancies as 'errors' or 'residuals' to highly valuable measures for model improvement. The field of data-driven engineering for dynamical systems has an opportunity to improve system characterization by disambiguating deterministic and random effects within model-measurement mismatch. In this paper, we formalize the mathematical infrastructure of discrepancy modeling for dynamical systems, highlighting the interplay and balance between deterministic and random effects. Specifically, we consider four data-driven modeling methods for identifying discrepancy models and provide two principled approaches for evaluating the utility and suitability of discrepancy modeling. We leverage recent mathematical advancements in data-driven model discovery and evaluate the interplay between $g(\mathbf{x}, t)$ and $\mathcal{N}(\mu, \sigma)$, showing how their relative sizes determine the ability to disambiguate deterministic from random or noisy effects. To encourage exploration and expansion of discrepancy modeling, we employ base coding packages for each model discovery method implementation [23–36]. We found that the performance characteristics of our suite of model discovery methods matched their documented performance for dynamical systems modeling. All the methods proposed (SINDY, DMD, GPR & NN) can be used successfully for both modeling the missing physics or residual. Thus, the evaluation of an appropriate method involves the *intent* of the user (*i.e.* interpretability of the discrepancy model) and the computational efficiency and robustness of the method for a given *quantity* and *quality* of data.


(a) Discrepancy modeling for improved characterization of system dynamics

(b) Two approaches for building a discrepancy model: model systematic residual and learn missing physics

Figure 1: **Top panel**: An approximate dynamical model $f(\cdot)$ provides estimates of system behavior, $\mathbf{x}(t)$. However, true behavior $\mathbf{y}_k(t)$ deviates from these estimates. This model-measurement mismatch, or *discrepancy*, contains both deterministic and random effects. The goal of discrepancy modeling is to learn a discrepancy model $G(\cdot)$ that disambiguates the missing deterministic effects and augments the approximate dynamics to improve system characterization, $\tilde{\mathbf{x}}$. **Bottom panel**: There are two approaches for discrepancy model: (i) modeling systematic state-space residual, *i.e.*, the systematic error between the approximate state space, $\mathbf{x}(t)$, and measurements, $\mathbf{y}_k(t)$, and (ii) learning missing deterministic physics from the dynamical error, *i.e.*, the error between the derivative of the measurements, $\dot{\mathbf{y}}_k$, and the approximate dynamics given the state measurements, $f(\mathbf{y}_k)$.

2 Methods for Data-Driven Modeling

In this section we briefly describe the suite of data-driven modeling methods used for learning discrepancy models. It should be noted that there are many variants and improvements that can be implemented for each method considered. However, this comparative study does not aim to optimized the ability of each technique to perform at its absolute best. Indeed, hyper-parameter tuning of each technique is problem specific and again depends upon the *quality* and *quantity* of data and the intent of its use. Thus our goal is to demonstrate the various possibilities and their appropriate uses.

2.1 Gaussian Process Regression (GPR)

Gaussian process regression (GPR) is a non-parametric supervised learning approach to regression that uses a Gaussian process prior for Bayesian inference [27]. Not only is GPR a powerful nonlinear interpolation tool, its inherent probabilistic architecture allows for uncertainty quantification of interpolated values. Consider the dataset $D\{(\mathbf{X}_i, \mathbf{Y}_i | i = 1, \ldots, n\}$ which is split into test and training subsets where $\mathbf{X}_i$ are individual observations and $\mathbf{Y}_i$ are observation labels. A Gaussian process $f(\mathbf{X})$ is defined by a mean function $m(\mathbf{X})$ and covariance function $k(\mathbf{X}, \mathbf{X}')$ as

$$\begin{aligned} m(\mathbf{X}) &= \mathbb{E}[f(\mathbf{X})] \\ k(\mathbf{X}, \mathbf{X}') &= \mathbb{E}[(f(\mathbf{X}) - m(\mathbf{X}))(f(\mathbf{X}') - m(\mathbf{X}'))] \end{aligned}$$

such that the Gaussian process is:

$$f(\mathbf{X}) \sim \mathcal{GP}(m(\mathbf{X}), k(\mathbf{X}, \mathbf{X}')). \tag{6}$$

GPR makes several critical assumptions about the measurement noise: (i) it is additive, (ii) it is *iid* (independent and identically distributed), and (iii) it is given by a Gaussian distribution $\epsilon \sim \mathcal{N}(0, \sigma^2)$ with mean zero and variance, σ. Kernel functions are used to calculate covariance and enforce assumptions about the data. Specifically, kernels are parameterized by hyperparameters controlling covariance characteristics (*e.g.* length-scales or periodicity) and are optimized by maximizing the log marginal likelihood during model selection. As the prior distribution is a Gaussian process $f(\mathbf{X}) \sim \mathcal{GP}$, the conditional distribution $f(\mathbf{X})|D_n$ is the posterior, i.e. the *predictive distribution*. Computing on a finite-sized data set and partitioning it into training outputs ($\mathbf{f}$) and test outputs ($\mathbf{f}_*$), the prior joint distribution is assumed to take the form:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*). \end{bmatrix}\right) \tag{7}$$

To find the posterior, or predictive, distribution, the joint prior distribution must be restricted, or conditioned, to contain functions that agree with the observations, $\mathbf{f}_*|X_*, X, \mathbf{f} \sim \mathcal{N}(\bar{\mathbf{f}_*}, \text{cov}(\mathbf{f}_*))$, which can be written in closed form [27]. An important note: GPR for prediction or parameter estimation is computationally expensive. Calculating the maximum likelihood requires finding the determinant and inverse of the covariance matrix, which has cubic computational complexity. Ultimately, GPR is about regressing to a Gaussian distribution and estimating the appropriate variances via (7). The mathematical details of GPR can be found here [27].

2.2 Dynamic Mode Decomposition (DMD)

Dynamic mode decomposition (DMD) is a modern system identification approach based on data-driven regression. DMD extracts spatio-temporal structures from time series data and learns a low-dimensional linear model describing the evolution of features that encode salient system behavior. Consider a dynamical system measured at evenly-spaced time points $\mathbf{t} = [t_1, t_2, \cdots, t_m]$.

From measurements, we construct a matrix of snapshots $\mathbf{X}(\mathbf{t}) = [\mathbf{x}_1(\mathbf{t})\ \mathbf{x}_2(\mathbf{t})\ \ \ldots\ \ \mathbf{x}_n(\mathbf{t})] \in \mathbb{R}^{m\times n}$. A discrete time linear representation of a system is assumed to take the standard form, $\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t$ such that the linear operator $\mathbf{A}$ progresses the state vector $\mathbf{x}_t$ forward in time. Mirroring this standard form, two snapshot matrices are defined as:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{m-1} \\ | & | & & | \end{bmatrix}, \tag{8}$$

$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix} \tag{9}$$

where $\mathbf{X}'$ is the time-shifted matrix of snapshots $\mathbf{X}$, *i.e.* $\mathbf{X}' = \mathbf{A}\mathbf{X}$. The *exact* DMD is the best fit linear mapping $\mathbf{A}$ between snapshot pairs $\mathbf{X}$ and $\mathbf{X}'$:

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F = \mathbf{X}'\mathbf{X}^\dagger \tag{10}$$

where $\|\cdot\|_F$ is the Frobenius norm and † is the Moore-Penrose pseudoinverse [26]. DMD exploits the singular value decomposition (SVD) to solve for:

$$\mathbf{A} = \mathbf{U}^*\mathbf{X}'\mathbf{V}\mathbf{\Sigma}^{-1}. \tag{11}$$

DMD exploits low-rank structure of high-dimensional systems, and thus projects $\mathbf{A}$ onto the first r modes of the principle components $\mathbf{U_r}$. This rank-r truncation of $\mathbf{X} \approx \mathbf{U_r}\mathbf{\Sigma_r}\mathbf{V_r^*}$ approximates the pseudo-inverse:

$$\tilde{\mathbf{A}} = \mathbf{U}_r^*\mathbf{A}\mathbf{U}_r = \mathbf{U}_r^*\mathbf{X}'\mathbf{V}_r\mathbf{\Sigma}_r^{-1}. \tag{12}$$

The eigendecomposition of $\tilde{\mathbf{A}}$ yields the eigenvalues and eigenvectors, $\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}$, which provides insight into underlying system properties such as growth modes and resonance frequencies [26]. The eigenvectors of $\mathbf{A}$ are the *DMD modes* $\mathbf{\Phi}$:

$$\mathbf{\Phi} = \mathbf{X}'\tilde{\mathbf{V}}\tilde{\mathbf{\Sigma}}^{\mathbf{-1}}\mathbf{W}. \tag{13}$$

Along with the mode amplitudes, $\mathbf{b} = \mathbf{\Phi}^\dagger\mathbf{x}_1$, the well known DMD solution takes the form

$$\mathbf{x}(t) = \sum_{i=1}^{r} \phi_i e^{\omega_i t} b_i = \mathbf{\Phi}\exp(\mathbf{\Omega}\mathbf{t})\mathbf{b}. \tag{14}$$

However, exact DMD is prone to biased errors resulting from noisy measurements, affecting model fit and forecasting stability [31]. Therefore, Askham and Kutz [32] introduced *optimized* DMD, which uses variable projection to perform nonlinear optimization for de-biasing model fitting in the presence of observation noise. More specifically, the variable projection method optimally computes nonlinear least squares exponential fitting for DMD:

$$\underset{\omega,\Phi_b}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{\Phi}\exp(\mathbf{\Omega}t)\mathbf{b}\|_F. \tag{15}$$

In this paper, we use optimized DMD, but generally refer to it as 'DMD'. Note: the rank of $\mathbf{A}$ cannot exceed the state dimension of $\mathbf{X}$, and DMD algorithms rely on the availability of full-state measurements, typically of high dimensions. When only partial observations or low-dimensional system measurements are available, it is helpful to build an augmented state vector that is 'lifted' into a higher dimension. For our study, we accomplish this via time-delay embedding, which also happens to result in an intrinsic coordinate system forming a Koopman-invariant subspace in which nonlinear dynamics appear linear [33].

2.3 Sparse identification of nonlinear dynamics (SINDy)

Sparse identification of nonlinear dynamics (SINDy) recovers parsimonious representations of the dynamics from measurement data by sparse regression to a library of candidate models [23, 25, 29]. Consider a nonlinear dynamical system measured at time points $\mathbf{t} = [t_1, t_2, \cdots, t_m]$. From measurements, we construct the matrix $\mathbf{X}(\mathbf{t}) = [\mathbf{x}_1(\mathbf{t}) \; \mathbf{x}_2(\mathbf{t}) \; \ldots \; \mathbf{x}_n(\mathbf{t})] \in \mathbb{R}^{m \times n}$. The method introduced in [29] seeks to identify $\mathbf{f}$ via sequential threshold least-squares, which is a proxy for the sparsifying zero-norm. The set of n state measurements are used to populate a library of candidate nonlinear terms $\boldsymbol{\Theta}(\mathbf{X}) = [\mathbf{1}^\top \; \mathbf{X}^\top \; (\mathbf{X} \otimes \mathbf{X})^\top \; \cdots \; \sin(\mathbf{X})^\top]$, where $\mathbf{x} \otimes \mathbf{y}$ defines the vector of all product combinations of the state components. Each candidate term should be unique, as a suitable library is crucial in the SINDy algorithm. A common strategy is to start with polynomials and increase the complexity of the library with other terms, such as trigonometric functions. Thus, a dynamical system can be re-written as:

$$\dot{\mathbf{X}} = \boldsymbol{\Theta}(\mathbf{X})\boldsymbol{\Xi}. \tag{16}$$

The time derivatives $\dot{\mathbf{X}}(\mathbf{t}) = [\dot{\mathbf{x}}_1(\mathbf{t}) \; \dot{\mathbf{x}}_2(\mathbf{t}) \; \ldots \; \dot{\mathbf{x}}_n(\mathbf{t})]$, if not measured directly, can be found via numerical differentiation and should be appropriately de-noised, if necessary [37]. The coefficients $\boldsymbol{\Xi}$ are the *sparse* weightings of the corresponding candidate library terms. Therefore, our regression relies on sparse regularization to enforce a parsimonious $\boldsymbol{\Xi}$ corresponding to the fewest nonlinear terms in our library that describe our dynamics well:

$$\boldsymbol{\Xi} = \text{arg}\min_{\hat{\boldsymbol{\Xi}}} \|\boldsymbol{\Theta}(\mathbf{X})\hat{\boldsymbol{\Xi}} - \dot{\mathbf{X}}\|_2 + \lambda\|\hat{\boldsymbol{\Xi}}\|_0 \tag{17}$$

Regressing to the zero-norm is often achieved by relaxing the one-norm. However, modern optimization frameworks are allowing for computationally tractable proxies for the zero-norm that are superior to the one-norm relaxation [24].

2.4 Neural Networks (NN)

Artificial neural networks, or simply *neural networks* (NN), are mathematical models inspired by biological neural networks. While there is a wide array of literature on NN [28, 38–44], we outline the basic concepts. NN learn a mapping between a set of input data and target outcomes. The middle, or hidden, layers form a compositional structure that optimizes the association between the training data set. The user designates the depth (number of hidden layers), the dimensionality (number of nodes) of each layer, and how each layer is connected.

Linearly, this means a NN optimizes over the compositional function to learn the neural network weights and biases matrices $\mathbf{A}_j$ between the k-hidden layers:

$$\underset{\mathbf{A}_k}{\text{argmin}}(f_M(\mathbf{A}_M, \ldots, f_2(\mathbf{A}_2, f_1(\mathbf{A}_1, \mathbf{x}))\ldots) + \lambda g(\mathbf{A}_j)), \tag{18}$$

where $\lambda g(\mathbf{A}_j)$ is included to provide and appropriate regularization for the solution. For example, a simple, single hidden layer ($k = 1$) NN is structured as:

$$\begin{aligned} \mathbf{x}^{(1)} &= \mathbf{A}_1\mathbf{x} \\ \mathbf{y} &= \mathbf{A}_2\mathbf{x}^{(1)}. \end{aligned} \tag{19}$$

Leveraging the compositional structure, a mapping is defined by:

$$\mathbf{y} = \mathbf{A}_2\mathbf{A}_2\mathbf{x}, \tag{20}$$

which generalizes to M layers

$$\mathbf{y} = \mathbf{A}_M \mathbf{A}_{M-1} \dots \mathbf{A}_2 \mathbf{A}_1 \mathbf{x}. \tag{21}$$

Nonlinear mappings are structured similarly. In this case, nonlinear activation functions connect hidden layers and are given by:

$$\begin{aligned} \mathbf{x}^{(1)} &= f_1(\mathbf{A}_1, \mathbf{x}) \\ \mathbf{y} &= f_2(\mathbf{A}_2, \mathbf{x}^{(1)}). \end{aligned} \tag{22}$$

Further, nonlinear activation functions, $f_j(\cdot)$, can differ between layers. Thus, nonlinear mapping between a given set of input and output data over M layers is structured as:

$$\mathbf{y} = f_M(\mathbf{A}_M, \dots, f_2(\mathbf{A}_2, f_1(\mathbf{A}_1, \mathbf{x})) \dots) = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}) \tag{23}$$

where $\mathbf{f}_{\boldsymbol{\theta}}(\cdot)$ represents the overall network structure with weights and biases $\boldsymbol{\theta}$. While often used for classification, NNs can be structured to learn the evolution of dynamical systems. NN for dynamical systems provides a flexible and powerful architecture for high-dimensional supervised learning of system behavior for future state predictions [45].

3 Discrepancy Modeling Framework

3.1 The Two Approaches

Discrepancy modeling aims to improve system characterization from data by disambiguating deterministic and random effects within the model-measurement mismatch. The approximate, or Platonic, model is augmented with a learned discrepancy model to improve dynamical system characterization. Indeed, there are two nuanced, yet distinct, means of building a discrepancy model: (i) discover a model for missing deterministic physics, or (ii) learn a model for the systematic state-space residual. In the first approach, the discrepancy dynamics are the dynamical error, *i.e.*, the error between the derivative of the measurements, $\dot{\mathbf{y}}_k$, and the approximate dynamics given the state measurements, $f(\mathbf{y}_k)$. Thus, learning a discrepancy model is akin to learning missing physics and is to be appended to the approximate dynamics. In the second approach, the discrepancy dynamics are the state-space residual, *i.e.*, the systematic error between the approximate state space, $\mathbf{x}$, and the measurements, $\mathbf{y}_k$. Thus, this discrepancy model acts as a correction to the approximate state-space solution.

3.2 Learning Missing Physics

In this approach, the discrepancy model recovers missing deterministic physics within the model-measurement mismatch to improve the known, approximate model. The discrepancy dynamics are the dynamical error between measurement derivatives and approximate dynamics, $\mathbf{E}_f(t) = \dot{\mathbf{y}}_k(t) - f(\mathbf{y}_k(t))$. The formulation begins with Eqns. (1-3). Using the suite of data-driven methods proposed, a discrepancy model $G(\mathbf{y}_k, t)$ is learned using the dynamical error calculated as:

$$\begin{aligned} \text{GPR}: &\quad \mathbf{E}_f(t) \sim \mathcal{GP}(m(\mathbf{y}_k), k(\mathbf{y}_k, \mathbf{y}'_k)) \\ \text{DMD}: &\quad \mathbf{E}_f(t) \approx \boldsymbol{\Phi}\ \text{diag}(\mathbf{b}) e^{\boldsymbol{\omega}\mathbf{t}}, \quad \mathbf{b} = \boldsymbol{\Phi}^\dagger \mathbf{y}_k \\ \text{SINDy}: &\quad \mathbf{E}_f(t) = \Theta(\mathbf{y}_k)\Xi \\ \text{NN}: &\quad \mathbf{E}_f(t) = f(\mathbf{y}_k) \end{aligned}$$

and is appended to the approximate dynamics:

$$\frac{d\tilde{\mathbf{x}}}{dt} = \tilde{F}(\tilde{\mathbf{x}}, t) = f(\tilde{\mathbf{x}}, t) + G(\tilde{\mathbf{x}}, t)$$

to minimize:

$$\tilde{\mathbf{E}}_f(t) = \dot{\mathbf{y}}_k(t) - \tilde{F}(\mathbf{y}_k(t)) \tag{24}$$

3.3 Modeling Systematic State-Space Residual

In this approach, the discrepancy model learns the time evolution of systematic state-space residual and corrects the approximate state-space solution. The discrepancy model framework begins with Eqns. (1-3). Using the suite of data-driven methods proposed, a discrepancy model $G(\mathbf{x}, t)$ is learned using the systematic residual calculated as:

$$\begin{aligned} \text{GPR}: \quad & \mathbf{E}(t) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \\ \text{DMD}: \quad & \mathbf{E}(t) \approx \mathbf{\Phi}\ \text{diag}(\mathbf{b}) e^{\boldsymbol{\omega} \mathbf{t}}, \quad \mathbf{b} = \mathbf{\Phi}^{\dagger}\mathbf{x} \\ \text{SINDy}: \quad & \mathbf{E}(t) = \Theta(\mathbf{x})\Xi \\ \text{NN}: \quad & \mathbf{E}(t) = f(\mathbf{x}) \end{aligned}$$

and corrects the approximate state-space solution:

$$\tilde{\mathbf{x}}(t) = \mathbf{x}(t) + G(\mathbf{x}, t)$$

to minimize:

$$\tilde{\mathbf{E}}(t) = \mathbf{y}_k(t) - \tilde{\mathbf{x}}(t).$$

4 Discrepancy Modeling Applications

Discrepancy modeling is both system- and situational-dependent; thus we evaluate the utility and suitability of each discrepancy modeling approach using a suite of four model discovery methods by comparing (i) discrepancy reconstruction error and (ii) augmented forecasting accuracy versus approximate and true dynamics. We further probe discrepancy modeling performance on increasingly complex systems and for increasing levels of noise.

4.1 Van der Pol Oscillator: A Simple Example

We began with a simple model and no noise. The data used in this example was generated using the Van der Pol oscillator:

$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x = 0 \tag{25}$$

using $x_0 = [0.1, 5]$, $t = [0, 50]$, and $\Delta t = 0.01$. We simulated Eqn. (25) to generate our Platonic or approximate dynamics. To this system, we added a small nonlinear term ϵx^3:

$$\frac{d^2x}{dt^2} - \mu(1 - x^2)\frac{dx}{dt} + x + \epsilon x^3 = 0 \tag{26}$$

Eqn. (26) was simulated to generate the true system behavior using $\epsilon = 0.01$. This ϵ-small nonlinearity represents the missing deterministic physics not captured in the approximate model. This ϵ cubic term added to the approximate dynamics perturbed the time evolution of Van der Pol, as seen in Fig. (2), while still maintaining salient characteristics associated with the oscillator.

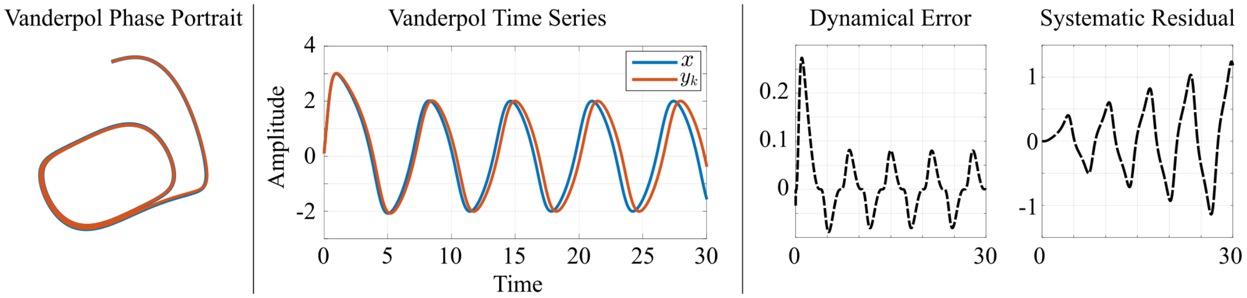


Figure 2: Van der Pol oscillator example (no noise) with and without a discrepancy. While the salient dynamical features are preserved as seen in the phase portrait (left panel), the time evolution (middle panel) diverges quickly with only an ϵ-small dynamical difference. The dynamical error and systematic residual (right panel) are plotted to demonstrate the two discrepancy types.

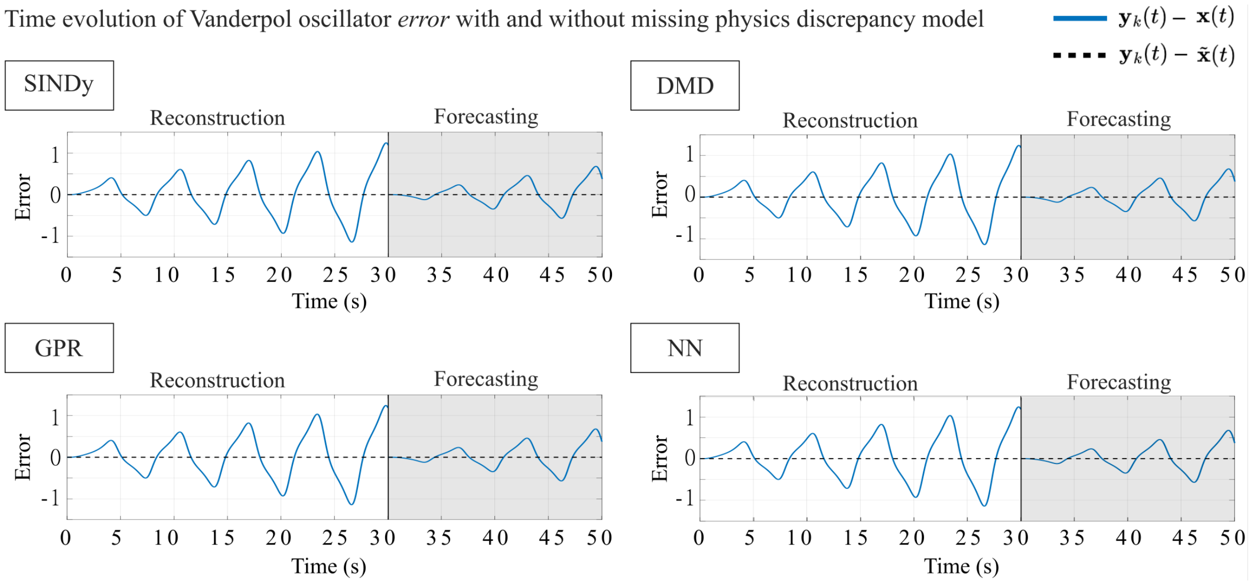


Figure 3: Remaining state-space error with and without a discrepancy model of the missing physics appended to the approximate Van der Pol oscillator model (no noise). The blue line shows the error without a discrepancy model, and the black dashed line shows the error with a discrepancy model recovering the missing physics. The suite of model discovery methods learned the missing physics within the discrepancy such that no error remained.

4.1.1 Learning Missing Physics

We first evaluated the ability of discrepancy modeling to recover missing physics for the Van der Pol oscillator. As seen in Fig (3), the suite of model discovery methods learned the missing physics within the discrepancy such that no error remained. Both the reconstruction and the forecasting errors (black dashed line) between the true and augmented models is zero for all model discovery methods. The discrepancy dynamics between the true and approximate models are denoted by the blue line. The Van der Pol oscillator has a parameter dictating the nonlinearity of its oscillations; we chose a mild level of nonlinearity to start. As the nonlinearity parameter increases, we would expect that *linear* model discovery methods (*e.g.*, DMD) would struggle to build an accurate discrepancy model and thus fail to fully recover missing physics.

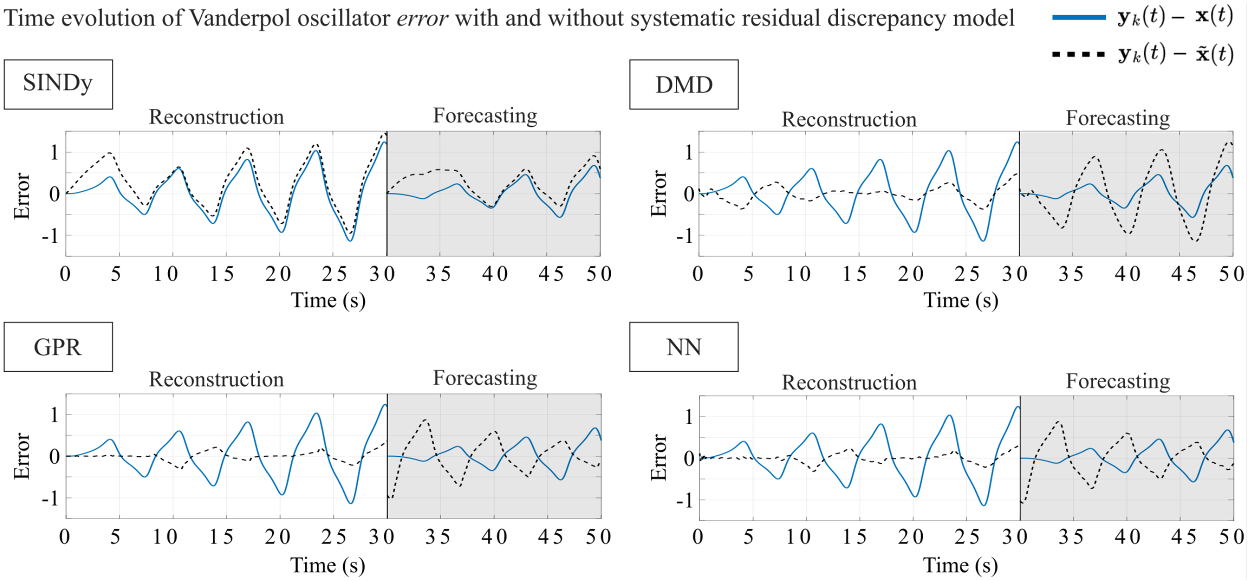


Figure 4: Remaining state-space error with and without a discrepancy model of the systematic residual to correct the approximate Van der Pol oscillator state-space solution (no noise). The blue line shows the error without a discrepancy model, and the black dashed line shows the error with a discrepancy model modeling the systematic residual. The suite of model discovery methods had various success in modeling the systematic residual, resulting in promising utility of discrepancy modeling for correcting state-space solutions.

4.1.2 State-Space Error

We next evaluated the ability of discrepancy modeling to learn the evolution of the systematic state-space residual for the Van der Pol oscillator. As seen in Fig. (4), the suite of model discovery methods had various success in modeling the time evolution of the systematic residual, thus demonstrating promising utility of discrepancy modeling for correcting state-space solutions. SINDy failed to learn the systematic residual, thus resulting in reconstruction and forecasting errors similar to if no discrepancy model was used. The inability for SINDy to model systematic residual is unsurprising, as it is formulated to recover dynamical terms (See Eqn. 16). DMD, GPR, and NN fared well in modeling the systematic residual, resulting in reduced error as compared to the approximate model; however, these methods struggled to correct forecasts of the approximate model. DMD was able to capture the salient features of the residual, yet the augmented solution appeared to be slightly time-shifted from the true solution, as seen by the brief zero-error within the reconstruction regime. Both GPR and NN learn promising discrepancy models of the systematic residual, as they both had zero error between the true and augmented solutions for the first ten seconds in the reconstruction region, but began to diverge, albeit minimally. However, both GPR and NN had large errors at the beginning of forecasting. We theorize this is due to a failure to appropriately initialize. Generally, both methods learn their training trajectories with high accuracy; often, large datasets with many trajectories, and thus many initial conditions, were used in model training. In our case, because only one training trajectory was used, *i.e.*, one set of initial conditions, GPR and NN were unable to extrapolate the expected systematic residual when initialized with a different initial condition for forecasting. We theorize this initialization error can be resolved to greatly reduce remaining error when using a systematic residual discrepancy model via: (i) increasing data *quantity* (number of trajectories, initial conditions, parameters) for

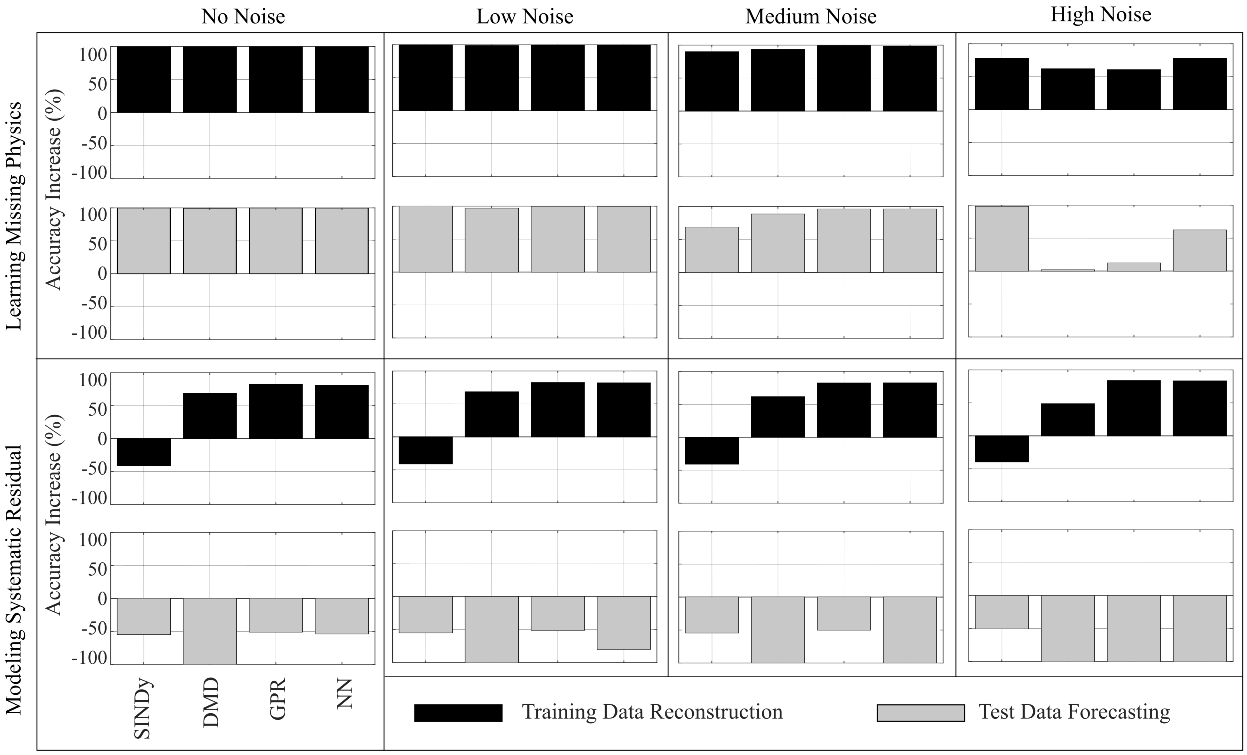


Figure 5: Increase in accuracy with discrepancy modeling augmentation in Van der Pol oscillator. Increase in % accuracy is calculated as the relative change between the root mean squared errors (RMSE) of the augmented and approximate solutions as compared to the true solution. Results are shown for no (0%), low (0.1%), medium (1%), and high (10%) noise levels. Discrepancy modeling can be successful in numerous ways and depends on user intent (*e.g.*, interpretability), data quality (*e.g.*, noise tolerance), and data quantity (*e.g.*, rapid evaluation vs. computational cost).

training, or (ii) sacrificing the first portion of the test data/new times series to update the trained model.

4.1.3 Adding Noise

We re-ran both discrepancy modeling approaches with the Van der Pol oscillator and increasing levels of Gaussian noise added to the 'true' Van der Pol dynamics to evaluate the effect of sensor measurement *quality*. Importantly, we assume any noise is due to both natural variability and system observation. We use noise level of $\sigma = [0.1\%,\ 1\%,\ 10\%]$. As seen in Fig. 5, discrepancy modeling can be successful in numerous ways. Therefore, implementation amounts to user intent for discrepancy modeling, as well as the quantity and quality of sensor measurements. For example, when using discrepancy modeling to learn missing physics of the Van der Pol oscillator, all methods reconstruct and forecast true dynamics successfully in the no, low, and medium noise regimes. Note in the high noise regime, accuracy will most likely increase for methods like SINDy (which has an accuracy increase but is non-sparse) and DMD with innovations from baseline code packages (*e.g.*, ensemble, bagging, culling) [34, 46, 47]. If the user's intent for discrepancy modeling is interpretability, for no/low/medium noise, SINDy and DMD perform well, with minimal data quantity requirements and small computational costs (See Fig. 11 for computational cost comparisons). Even if the user has no interest in interpretability, SINDy and DMD

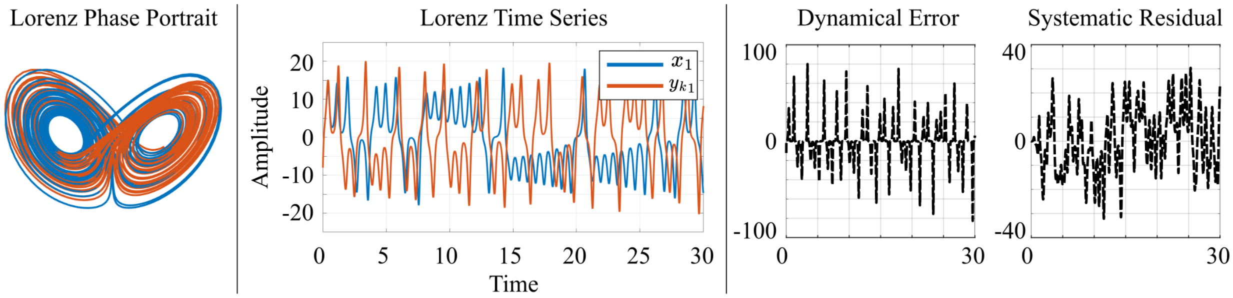


Figure 6: Lorenz attractor example (no noise) with and without a discrepancy. While the salient dynamical features are preserved as seen in the phase portrait (left panel), the time evolution (middle panel) bifurcates quickly with only an ϵ-small dynamical difference. Chaotic dynamical systems like the Lorenz attractor are particularly sensitive to small errors in system dynamics. The dynamical error and systematic residual (right panel) are plotted to demonstrate the two types of discrepancies.

perform comparably to GPR and NN (methods that have increased data quantity requirements and greater computational costs). On the other hand, if the user's intent is to only reduce error, correcting the state-space solution with a discrepancy model of the systematic residual is feasible with DMD, GPR, and NN. While forecasting errors appear abysmal, utility of these methods for forecasting may increase with better initialization, as discussed above.

4.2 Lorenz Attractor: Adding Chaos

We extended our discrepancy modeling analyses to a more complex canonical system. We simulated the Lorenz attractor, which has increasingly complex dynamical features like chaos:

$$\frac{dx}{dt} = \sigma(y - x) \qquad \frac{dy}{dt} = x(\rho - z) - y \qquad \frac{dz}{dt} = xy - \beta z \tag{27}$$

with the oft-used parameters, $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$, as well as $t = [0, 50]$ and $\Delta t = 0.01$, to generate our Platonic or approximate dynamics. To this system, we again added an ϵ-small nonlinearity:

$$\frac{dx}{dt} = \sigma(y - x) + \epsilon x^3 \qquad \frac{dy}{dt} = x(\rho - z) - y \qquad \frac{dz}{dt} = xy - \beta z \tag{28}$$

and simulated to generate the true system behavior using $\epsilon = 0.01$. This ϵ cubic term added to the approximate dynamics perturbed the time evolution of Lorenz, as seen in Fig. 6, while still exhibiting salient characteristics associated with the attractor and maintaining dynamical stability.

4.2.1 Learning Missing Physics

We evaluated the ability of discrepancy modeling to recover missing physics for the Lorenz attractor. As seen in Fig. 7, the suite of model discovery methods had various success in learning the missing physics for Lorenz. SINDy, GPR, and NN did well at learning the missing physics, especially considering the sensitivity of chaotic systems to small dynamic deviations. These methods were able to reconstruct the first five seconds of the true Lorenz solution, as denoted by the zero-error in the reconstruction region (black dashed line). In the forecasting region, SINDy had zero-error briefly but quickly became erroneous; this was most likely due to a small parameter deviation from the true dynamics that allowed the augmented dynamics to diverge from the true

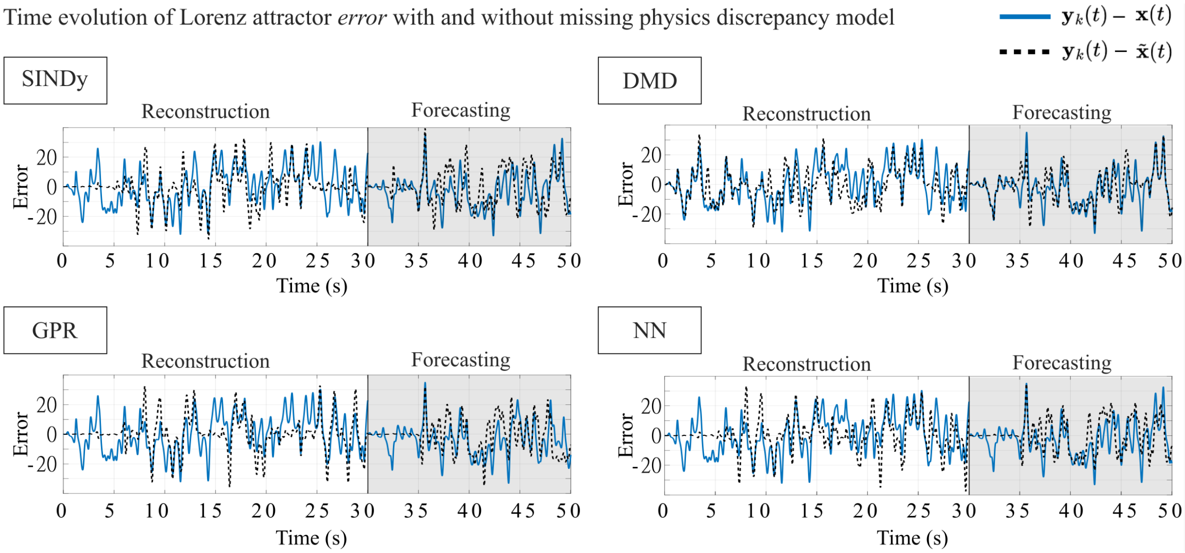


Figure 7: Remaining state-space error with and without a discrepancy model of the missing physics appended to the approximate Lorenz attractor dynamics (no noise). The blue line shows the error without a discrepancy model, and the black dashed line shows the error with a discrepancy model recovering the missing physics.

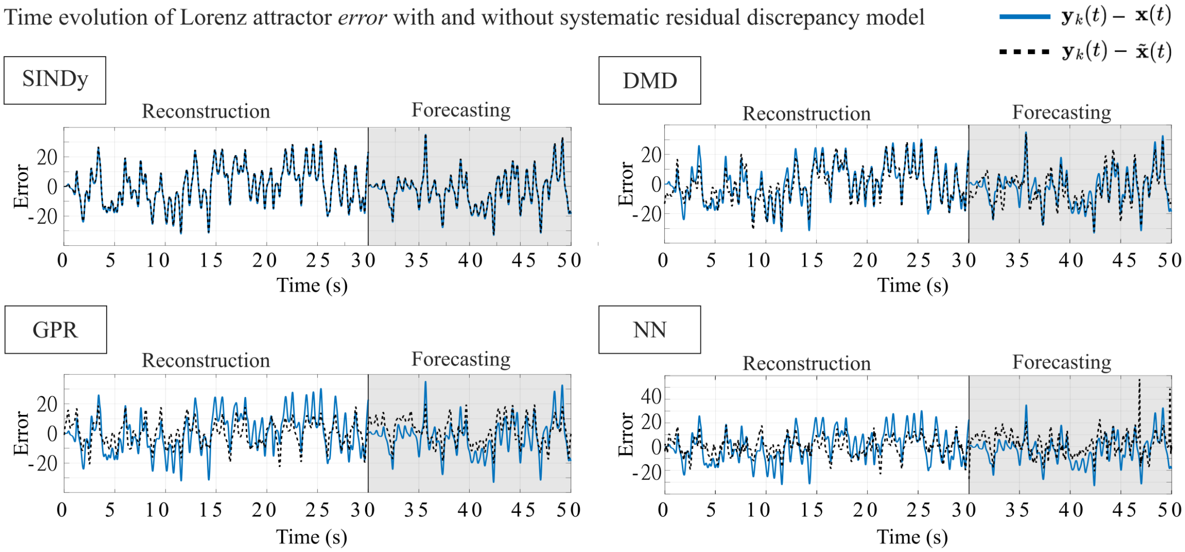


Figure 8: Remaining state-space error with and without a discrepancy model of the systematic residual to correct the approximate Lorenz attractor state-space solution (no noise). The blue line shows the error without a discrepancy model, and the black dashed line shows the error with a discrepancy model modeling the systematic residual. All model discovery methods struggled to learn the time evolution of the systematic residual.

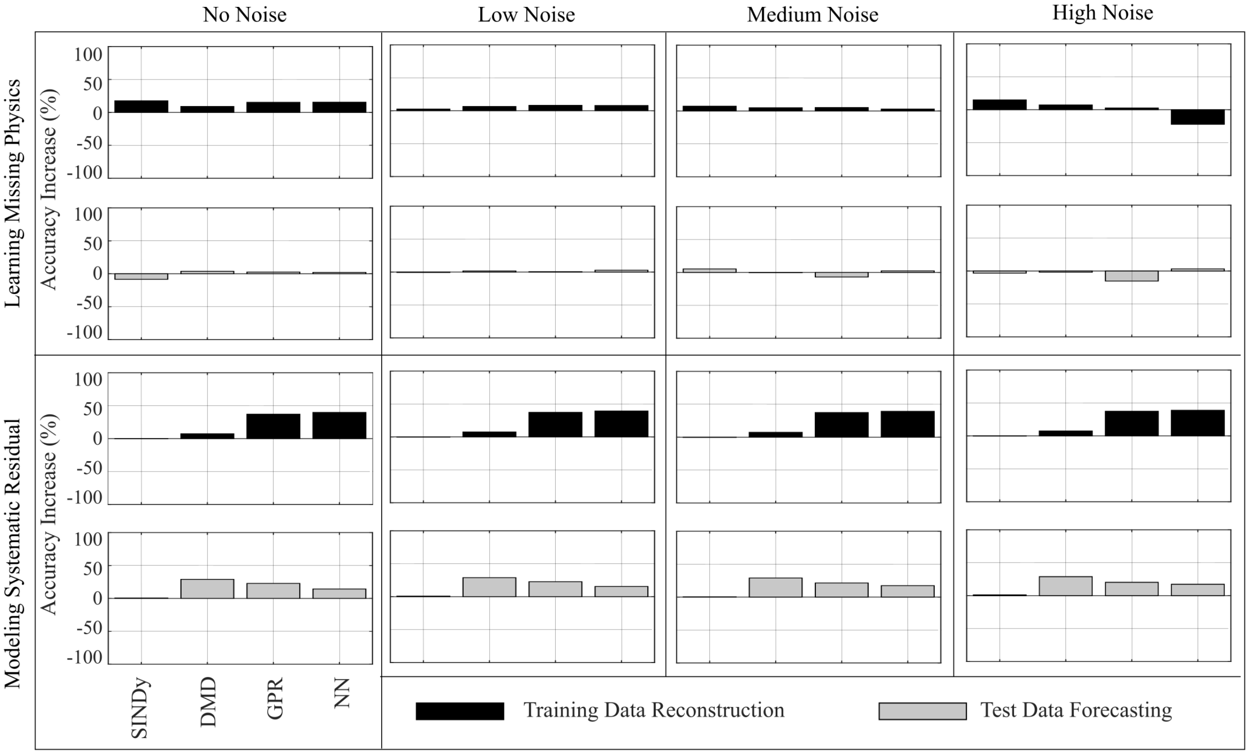


Figure 9: Increase in accuracy with discrepancy modeling augmentation in Lorenz attractor. Increase in % accuracy is calculated as the root mean squared error (RMSE) between true and augmented state space solutions for reconstruction of the training region and forecasting in the test region. Results are shown for no (0%), low (0.1%), medium (1%), and high (10%) levels of noise. Discrepancy modeling shows promise for overcoming model-measurement mismatch, even for nonlinear and chaotic systems such as the Lorenz attractor.

dynamics. Both GPR and NN were able to forecast the first five seconds of the true dynamics. In both the reconstruction and forecasting regions, the jump in non-zero error (blue line) corresponds with the attractor jump of the Lorenz dynamics. Unsurprisingly, DMD — a linear model discovery method — fails to learn the missing physics for the Lorenz attractor example. Innovations from DMD’s base code package have addressed some of the dynamical challenges due to nonlinearity and chaos, and continued innovation may allow for improved learning of missing physics [33].

4.2.2 State-Space Error

We next evaluated the ability of discrepancy modeling to learn the evolution of the systematic state-space residual for the Lorenz attractor. As seen in Fig. 8, the suite of model discovery methods struggled to model the time evolution of the systematic residual. Similar to Van der Pol results above, SINDy was unable to learn a discrepancy model the systematic error for the Lorenz attractor; the augmented state space showed no change in time series solution as compared to the approximate state space. Similarly, DMD showed minimal impact on the approximate state space solution when corrected using the learned systematic residual discrepancy model. GPR and NN exhibited reduced error in the augmented state space solutions as compared to the approximate; however, it does not appear that a discrepancy model of the systematic residual alone can recover the true Lorenz attractor solution. Using discrepancy modeling to learn the system-

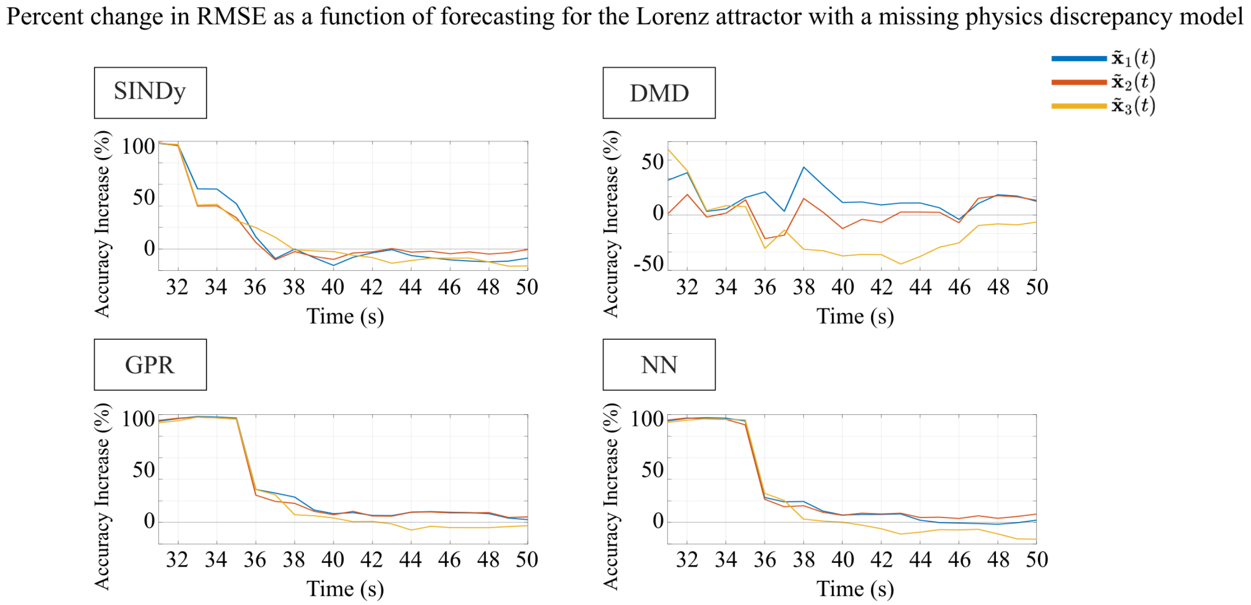


Figure 10: Percent change in RMSE as a function of forecasting. The discrepancy model for learning missing physics in the Lorenz attractor example (no noise) demonstrates the relationship between the accuracy increase and the forecasting window; here, we see a decrease in accuracy with discrepancy model augmentation as the forecasting window is extended and settle near zero percent change from the approximate model. Note: the drop in forecasting RMSE for NN, GPR and arguably the SINDy discrepancy models corresponds with the first attractor ’jump’ of the Lorenz. Predicting Lorenz bifurcations continues to be a challenging task.

atic residual may benefit from improved data quantity, as well as data assimilation techniques to combat known challenges with chaotic deterministic systems [48–50].

4.2.3 Adding Noise

We re-ran both discrepancy modeling approaches with the Lorenz attractor and increasing levels of Gaussian noise added to the ’true’ Lorenz dynamics to evaluate the effect of sensor measurement *quality*. Importantly, we assume any noise is due to both natural variability and system observation. We use noise level of $\sigma = [0.1\%,\ 1\%,\ 10\%]$. As seen in Fig. 9, discrepancy modeling shows promise for overcoming model-measurement mismatch, even for highly nonlinear, chaotic systems such as the Lorenz attractor. Firstly, despite seeing an increase in accuracy as compared to the approximate model in learning a discrepancy model of the systematic residual, the model discovery method were unable to reliably predict the attractor ’jump’, which is a known challenge for data-driven modeling of chaotic dynamical systems [33]. Secondly, we noticed a relationship between the accuracy increase and the forecasting window, as seen in Fig. 10. While discrepancy modeling appears to have little impact on reducing the model-measurement mismatch, the percent change in error in Fig. 9 was computed over the entire forecasting window. However, we see that percent change in root mean squared error (RMSE) is a function of forecasting. Accuracy increase with discrepancy model augmentation begins near 100% and decreases as the forecasting window is extended, eventually settling near zero percent change from the approximate model. Note the drop in RMSE around five seconds of forecasting for NN, GPR, and arguably the SINDy discrepancy models corresponds with the first attractor ’jump’ of the Lorenz.

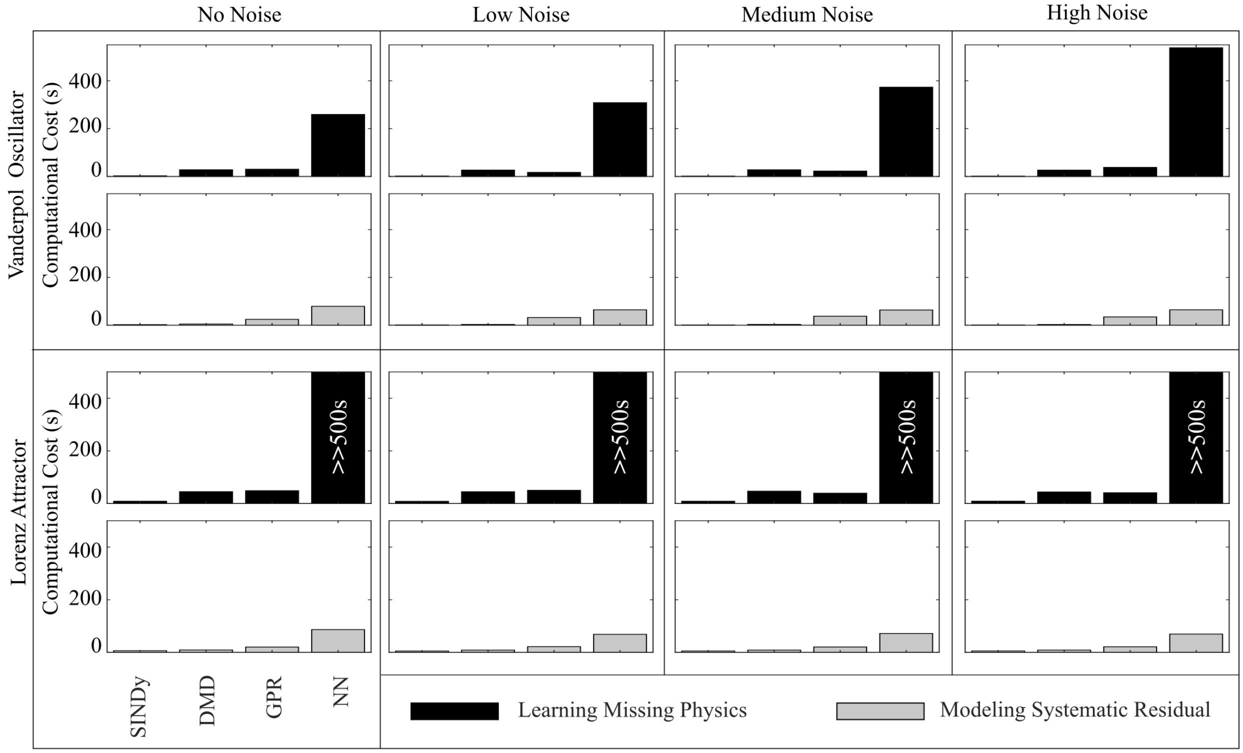


Figure 11: Computational costs (seconds) corresponding to executing the discrepancy modeling framework in MATLAB. Comparison across both approaches for each of the suite of model discovery methods. Cost is includes data generation for approximate and true models, computation of discrepancy dynamics, and reconstruction and forecasting of augmented model. The computational cost for the Van der Pol oscillator increases as noise increases for all model discover methods. SINDy has the lowest computational cost, followed by DMD, GPR, and NN, for both discrepancy modeling approaches. Learning missing physics with a NN has a noteably higher computational cost. This occurs because of how the discrepancy model is appended to the approximate dynamical model; the discrepancy dynamics are computed at each time step in MATLAB's ODE45 function, which greatly increases computational cost. These trends hold for the Lorenz attractor. Note: the time to execute the 'learning missing physics' script using the NN was much higher than 500 seconds, and peaked around 2100 seconds.

4.3 Burgers' Equation: Spatio-temporal Nonlinearity

Finally, we evaluate discrepancy modeling with a canonical system in partial differential equations. We applied the findings from our previous examples from the Van der Pol oscillator and the Lorenz attractor to inform our implementation of discrepancy modeling for the Burgers' equation. In this 'scenario', our intent for discrepancy modeling is for rapid evaluation with interpretability. Additionally, we have good data quality (no noise) and low data quantity (one set of spatio-temporal snapshots). Therefore, we chose to use DMD to learn a discrepancy model of the missing physics. We simulated Burgers' equation, which contains increasingly complex dynamical features, such as spatio-temporal nonlinearity:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{29}$$

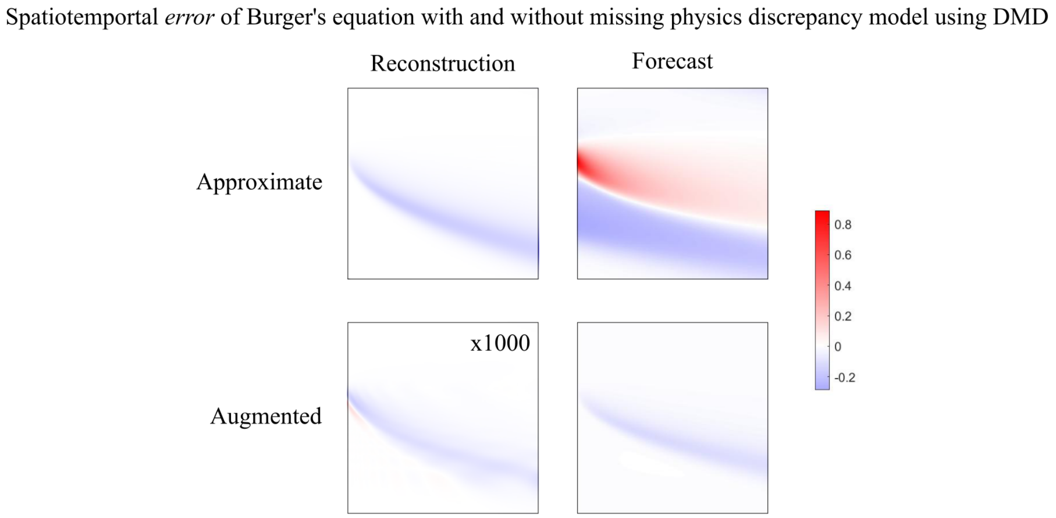


Figure 12: Remaining error with and without a discrepancy model of the missing physics appended to the approximate Burgers' dynamics (no noise). No color (white) represents zero error as compared to the true system. Both red and blue denote non-zero error; the different colors only distinguish positive and negative error, respectively. Note that the augmented reconstruction error is multiplied by 1000. The missing physics discrepancy model using DMD reconstructed true spatiotemporal dynamics with virtually zero remaining error and greatly diminished the error during forecasting. These results follow the same trends as seen in the Van der Pol example.

using $\nu = 0.1$ $t = [0, 50]$ and $\Delta t = 0.01$, to generate our Platonic or approximate dynamics. To this system, we again added an ϵ-small nonlinearity:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \epsilon u^3 = \nu\frac{\partial^2 u}{\partial x^2} \tag{30}$$

and simulate to generate the 'true' system behavior using $\epsilon = 0.01$. To create the data matrix $x \in \mathcal{R}^{1xn}$, Eqns. 29 and 30 were evaluated on a gridspace comprising $n = 256$ equally spaced spatial points and initiated using $u_0 = e^{(x+2)^2}$.

As seen in Fig. 12, DMD was successful in learning a discrepancy model recovering the missing physics in Burgers' equation. We note that trends from our non-chaotic ordinary differential equation example (Van der Pol) hold for this partial differential equation example (Burgers).

5 Conclusions and Guidelines for Discrepancy Modeling (Framework)

In conclusion, discrepancy modeling for learning missing physics, modeling systematic residuals, and disambiguating between deterministic and random effects emerged as an important framework for principled investigation of model-measurement mismatch. By leveraging improved observations to automate the process of learning better models, we can improve the characterization of underlying system dynamics. The two discrepancy modeling approaches introduced are distinct, yet nuanced: learning missing physics improves the underlying deterministic model and modeling systematic error corrects the known Platonic model. While an abundance of data-driven modeling methods exist, we adapted a suite of four model discovery methods to demonstrate the mathematical implications of discrepancy modeling. Each combination of approach and method for discrepancy modeling demonstrates the need for clearly defined goals and an understanding of constraints imposed by data collection. Further, by taking a deeper look at the mathematical implementations of each data-driven modeling method, we can better anticipate the utility and suitability of learning discrepancy models. Indeed, certain model discovery methods are more appropriate than others for each discrepancy modeling approach. It is important for the researcher to evaluate their *intent* for discrepancy modeling, as well as understanding sensor constraints (data *quality* and *quantity*) when pursuing discrepancy modeling to improve their dynamical model.

Demonstrated in this manuscript is discrepancy modeling for a number of canonical spatial and/or temporal systems. For each example system, we evaluate the reconstruction error and forecasting accuracy. In particular, we vary the signal-to-noise ratio to demonstrate the impact of random fluctuations on the ability to disambiguate between deterministic and random effects. An important implication of this work is that there is no ’silver bullet’ to automatically improve one’s underlying deterministic model. In fact, in certain cases when two opposing priorities cannot be reconciled, not using discrepancy modeling may be most appropriate. Further, this work demonstrates the limitations of discrepancy modeling as a function of sensor noise constraints; deterministic effects may exist within the model-measurement mismatch, but if random effects dominate, learning a discrepancy model of missing physics will be impossible and require improved sensor technology.

We summarize the the effects of the various discrepancy modeling paradigms:

(i) **SINDy:** While SINDy requires more training data and a high signal-to-noise ratio, it results in the recovery of parsimonious dynamics and thus is agnostic to nonlinearity strength in the discrepancy dynamics. SINDy has minimal computational costs and provides immense opportunity to improve engineering design through its interpretable approach to learning the governing dynamics. Additionally, SINDy provides the architecture to evaluate discrepancy model capability and parameter dependence of the learned discrepancy dynamics.

(ii) **Dynamic Mode Decomposition:** DMD’s strength is in its interpretable approach to rapid model evaluation. Further, DMD is a good discrepancy modeling method when the signal-to-noise is too low for SINDy and is not dominated by random fluctuations. As DMD is a linear data-driven modeling approach, a DMD discrepancy model may be sensitive to the nonlinearity strength of discrepancy dynamics.

(iii) **Gaussian process regression:** GPR is a powerful non-parametric Bayesian approach to discrepancy modeling in no/low/medium noise regime and lower data quantity requirements. It performed as well as — or, in certain cases, better than — a NN, along with lower associated computational costs.

(iv) **Neural networks:** NNs — a parametric modeling approach — shine when data in capturing complicated functions and generalizing to non-local behavior. If one can afford it, data assimilation for correcting systematic state space residual, for example with a GPR or NN, will be highly effective. For example, in chaotic systems such as the Lorenz, data assimilation will provide a consistent state-space correction versus modeling systematic error without feedback.

Remark: It should be noted that each architecture (i)-(iv) used can be enhanced and improved. For instance, both SINDy and DMD have recently been greatly improved by statistical bagging methods [46, 47]. Thus this comparative study does not claim to optimize the ability of each technique to perform at its absolute best. Indeed, hyper-parameter tuning of each technique is problem specific and again depends upon the *quality* and *quantity* of data and the intent of its use. Rather, our goal is to demonstrate the various possibilities and their appropriate uses. It is evident from the study that each method considered has strengths and weaknesses that are appropriate to consider depending upon data and intent.

Importantly, as data-driven modeling continues to gain momentum, it is imperative that researchers utilize domain knowledge (*e.g.*, first principles physics) to model complex systems. However, in all disciplines, model-measurement mismatch exists. The lack of investigation into the residual — due to the assumption that the mismatch or error is caused by random effects — leaves the missed opportunity to resolve model-measurement mismatch, disambiguate deterministic effects, and improve the underlying model. Discrepancy modeling provides a principled framework for both learning missing physics and modeling systematic error. Important considerations not explicitly evaluated in this manuscript include other dynamical factors of discrepancy model identification (*e.g.*, nonlinear magnitude of missing deterministic effects, quantity of data, latent space dynamics, etc). Future areas of investigation include adding control to the discrepancy modeling framework and combining model discovery methods for simultaneously learning discrepancy models of both deterministic and stochastic effects.

Acknowledgements

We are especially grateful to Kadierdan Kaheman and Steven Brunton for discussions related to discrepancy modeling. MRE acknowledges support from the NSF under award GRFP DGE-1762114. JNK acknowledges funding from the National Science Foundation AI Institute in Dynamic Systems grant number 2112085.

References

[1] Chia-Ch'iao Lin and Lee A Segel. *Mathematics applied to deterministic problems in the natural sciences*. SIAM, 1988.

[2] Carl M Bender and Steven A Orszag. *Advanced mathematical methods for scientists and engineers I: Asymptotic methods and perturbation theory*. Springer Science & Business Media, 2013.

[3] J Nathan Kutz. Advanced differential equations: Asymptotics & perturbations. *arXiv preprint arXiv:2012.14591*, 2020.

[4] Jared L Callaham, James V Koch, Bingni W Brunton, J Nathan Kutz, and Steven L Brunton. Learning dominant physical processes with data-driven balance models. *Nature communications*, 12(1):1–10, 2021.

[5] Joseph Bakarji, Jared Callaham, Steven L Brunton, and J Nathan Kutz. Dimensionally consistent learning with buckingham pi. *arXiv preprint arXiv:2202.04643*, 2022.

[6] Matteo Saveriano, Yuchao Yin, Pietro Falco, and Dongheui Lee. Data-efficient control policy search using residual dynamics learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4709–4715. IEEE, 2017.

[7] John Harlim, Shixiao W Jiang, Senwei Liang, and Haizhao Yang. Machine learning for prediction with missing dynamics. *Journal of Computational Physics*, 428:109922, 2021.

[8] Alban Farchi, Patrick Laloyaux, Massimo Bonavita, and Marc Bocquet. Using machine learning to correct model error in data assimilation and forecast applications. *Quarterly Journal of the Royal Meteorological Society*, 147(739):3067–3084, 2021.

[9] Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural lander: Stable drone landing control using learned dynamics. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9784–9790. IEEE, 2019.

[10] Matthew E Levine and Andrew M Stuart. A framework for machine learning of model error in dynamical systems. *arXiv preprint arXiv:2107.06658*, 2021.

[11] Kadierdan Kaheman, Eurika Kaiser, Benjamin Strom, J Nathan Kutz, and Steven L Brunton. Learning discrepancy models from experimental data. *arXiv preprint arXiv:1909.08574*, 2019.

[12] Cx K Batchelor and GK Batchelor. *An introduction to fluid dynamics*. Cambridge university press, 2000.

[13] Robert M Wald. *General relativity*. University of Chicago press, 2010.

[14] Brian M de Silva, David M Higdon, Steven L Brunton, and J Nathan Kutz. Discovery of physics from data: universal laws and discrepancies. *Frontiers in artificial intelligence*, 3:25, 2020.

[15] Philip R Bevington and D Keith Robinson. Data reduction and error analysis. *McGraw-Hill, New York*, 2003.

[16] John Taylor. *Introduction to error analysis, the study of uncertainties in physical measurements*. 1997.

[17] R Dennis Cook and Sanford Weisberg. *Residuals and influence in regression*. New York: Chapman and Hall, 1982.

[18] David R Cox and E Joyce Snell. A general definition of residuals. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(2):248–265, 1968.

[19] George EP Box and David A Pierce. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526, 1970.

[20] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[21] Kody Law, Andrew Stuart, and Kostas Zygalakis. Data assimilation. *Cham, Switzerland: Springer*, 214, 2015.

[22] Andrew C Miller, Nicholas J Foti, and Emily B Fox. Breiman's two cultures: You don't have to choose sides. *Observational Studies*, 7(1):161–169, 2021.

[23] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[24] Kathleen Champion, Peng Zheng, Aleksandr Y Aravkin, Steven L Brunton, and J Nathan Kutz. A unified sparse optimization framework to learn parsimonious physics-informed models from data. *IEEE Access*, 8:169259–169271, 2020.

[25] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[26] Jonathan H Tu, Clarence W Rowley, Dirk M Luchtenburg, Steven L Brunton, and J Nathan Kutz. On dynamic mode decomposition: Theory and applications. *arXiv preprint arXiv:1312.0041*, 2013.

[27] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[28] Kumpati S Narendra and Kannan Parthasarathy. Neural networks and dynamical systems. *International Journal of Approximate Reasoning*, 6(2):109–131, 1992.

[29] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[30] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

[31] Shervin Bagheri. Effects of weak noise on oscillating flows: Linking quality factor, floquet modes, and koopman spectrum. *Physics of Fluids*, 26(9):094104, 2014.

[32] Travis Askham and J Nathan Kutz. Variable projection methods for an optimized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 17(1):380–416, 2018.

[33] Steven L Brunton, Bingni W Brunton, Joshua L Proctor, Eurika Kaiser, and J Nathan Kutz. Chaos as an intermittently forced linear system. *Nature communications*, 8(1):1–9, 2017.

[34] Brian de Silva, Kathleen Champion, Markus Quade, Jean-Christophe Loiseau, J Kutz, and Steven Brunton. Pysindy: A python package for the sparse identification of nonlinear dynamical systems from data. 2020.

[35] Laura P Swiler, Mamikon Gulian, Ari L Frankel, Cosmin Safta, and John D Jakeman. A survey of constrained gaussian process regression: Approaches and implementation challenges. *Journal of Machine Learning for Modeling and Computing*, 1(2), 2020.

[36] Kadierdan Kaheman, Steven Brunton, and J Nathan Kutz. Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data. *Machine Learning: Science and Technology*, 2022.

[37] Floris Van Van Breugel, J Nathan Kutz, and Bingni W Brunton. Numerical differentiation of noisy data: A unifying multi-objective optimization framework. *IEEE Access*, 8:196865–196877, 2020.

[38] Raul González-García, Ramiro Rico-Martìnez, and Ioannis G Kevrekidis. Identification of distributed parameter systems: A neural net based approach. *Computers & chemical engineering*, 22:S965–S968, 1998.

[39] K Krischer, R Rico-Martínez, IG Kevrekidis, HH Rotermund, G Ertl, and JL Hudson. Model identification of a spatiotemporally varying catalytic reaction. *AIChE Journal*, 39(1):89–98, 1993.

[40] R Rico-Martinez, JS Anderson, and IG Kevrekidis. Continuous-time nonlinear signal processing: a neural network based approach for gray box identification. In *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, pages 596–605. IEEE, 1994.

[41] Isaac E Lagaris, Aristidis Likas, and Dimitrios I Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5):987–1000, 1998.

[42] Arka Daw, Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2017.

[43] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

[44] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[45] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[46] Diya Sashidhar and J Nathan Kutz. Bagging, optimized dynamic mode decomposition (bop-dmd) for robust, stable forecasting with spatial and temporal uncertainty-quantification. *arXiv preprint arXiv:2107.10878*, 2021.

[47] Urban Fasel, J Nathan Kutz, Bingni W Brunton, and Steven L Brunton. Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *arXiv preprint arXiv:2111.10992*, 2021.

[48] Geir Evensen. Advanced data assimilation for strongly nonlinear dynamics. *Monthly weather review*, 125(6):1342–1354, 1997.

[49] Pierre Gauthier. Chaos and quadri-dimensional data assimilation: A study based on the lorenz model. *Tellus A: Dynamic Meteorology and Oceanography*, 44(1):2–17, 1992.

[50] Robert N Miller, Michael Ghil, and Francois Gauthiez. Advanced data assimilation in strongly nonlinear dynamical systems. *Journal of Atmospheric Sciences*, 51(8):1037–1056, 1994.