Machine Learning and Data Assimilation for Blending Incomplete Models and Noisy Data

Thesis by
Matthew Emanuel Levine

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2023
Defended May 3, 2023

ACKNOWLEDGEMENTS

The work of MEL and AMS was supported by NIH RO1 LM012734 “Mechanistic Machine Learning.” MEL was also supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1745301. The computations presented here were conducted in the Resnick High Performance Center, a facility supported by Resnick Sustainability Institute at the California Institute of Technology.

In addition to financial support, I received an immense wealth of social and academic support from my friends, family, and the Caltech community. In particular, I wish to thank my advisor, Andrew Stuart, for his guidance, collaborative efforts, generosity, and kindness throughout my PhD. I am also grateful to my committee members, Yisong Yue, Houman Owhadi, and Katie Bouman, for their invaluable feedback on my work.

I also wish to thank my many collaborators and friends who have collectively brought irreplaceable joy to my graduate work: David Albers, Miguel Aparicio, Ricardo Baptista, Dima Burov, Edoardo Calvello, Yifan Chen, Jana de Wiljes, Oliver Dunbar, Michael Elowitz, Benjamin Emert, Emily Fox, Ricardo Garcia, Elia Gorokhovsky, Georg Gottwald, Franca Hoffman, Bamdad Hosseini, George Hripcsak, Daniel Huang, Ivan D. Jimenez Rodriguez, Nikola Kovachki, Vincent Lemaire, Zongyi Li, Haakon Ludvig, Lena Mamykina, Krithika Manohar, Elliott Mueller, Jacob Parres-Gold, Nicholas Nelsen, Elizabeth Qian, Sebastian Reich, Max Saccone, Tapio Schneider, Anish Senapati, Elnaz Seylabi, Jiaxin Shi, Melike Sirlanci, Margaret Trautner, Alex Wang, Robert Webber, Jinlong Wu, and Yisong Yue.

I am also deeply grateful to David Albers and Juliana Dias for taking me as their ward in beautiful Boulder, Colorado during a difficult time of global pandemic and stalled research progress. Finally, I wish to thank my mentors, teachers, coaches, friends, and family (especially my parents and sister) for setting me up for success in graduate school and supporting me through the inevitable challenges that arose along the way.

ABSTRACT

The prediction and inference of dynamical systems is of widespread interest across scientific and engineering disciplines. Data assimilation (DA) offers a well-established and successful paradigm for blending such models with noisy observational data. However, traditional DA-based inference often fails when available data are insufficiently informative. Chapter 2 copes with this challenge by introducing constraints into Ensemble Kalman Filtering, which is shown to improve forecasting of glucose dynamics in real patient-level clinical data. Chapter 3 addresses this identifiability challenge by instead developing a simplified, reduced-order stochastic model for glucose dynamics that is more easily identified from patient data. Despite these successes, the forecasting performance of the methods are fundamentally limited by the fidelity of the employed model, which is often not fully understood *a priori*.

Chapter 4 presents a general picture of how noisy, partially-observed time-series data can be used to learn flexible (e.g., neural network-based) corrections to a pre-specified mechanistic model. In Chapter 5, the proposed methodology is then validated in simulated settings for glucose-insulin models. Chapter 6 provides further perspective on learning flexible model corrections, comparing approaches that use i) gradient-based or gradient-free optimization, ii) temporal or time-averaged data, iii) different model parameterizations, iv) deterministic and stochastic corrections, and v) physical conservation laws to constrain inference.

Chapter 7 studies how these perspectives on machine learning and dynamical systems can help us understand the roles of biochemical networks. In particular, it considers protein dimerization networks from the lens of approximation theory and evaluates how the equilibria of these networks can be fine-tuned to perform a variety of biological computations.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] David J. Albers, Melike Sirlanci, Matthew E. Levine, Jan Classen, Caroline Der Nigoghossian, and George Hripcsak. “Interpretable Forecasting of Physiology in the ICU Using Constrained Data Assimilation and Electronic Health Record Data”. In: *Journal of Biomedical Informatics (In review)* (2023). arXiv: 2305.06513 [stat.AP].

[2] M. Sirlanci, M. E. Levine, C. C. Low Wang, D. J. Albers, and A. M. Stuart. “A Simple Modeling Framework For Prediction In The Human Glucose-Insulin System”. In: *Chaos (To appear)* (2023). arXiv: 1910.14193 [q-bio.QM].

[3] Ke Alexander Wang, Matthew E. Levine, Jiaxin Shi, and Emily B. Fox. “Learning Absorption Rates in Glucose-Insulin Dynamics from Meal Covariates”. In: *NeurIPS 2022 Workshop on Learning from Time Series for Health*. 2023. arXiv: 2304.14300 [cs.LG].

[4] ME Levine and AM Stuart. “A Framework for Machine Learning of Model Error in Dynamical Systems”. In: *Communications of the American Mathematical Society* 2.07 (2022), pp. 283–344. ISSN: 2692-3688. DOI: 10.1090/cams/10.

[5] David J. Albers, Paul-Adrien Blancquart, Matthew E. Levine, Elnaz Esmaeilzadeh Seylabi, and Andrew Stuart. “Ensemble Kalman methods with constraints”. en. In: *Inverse Problems* 35.9 (Aug. 2019). Publisher: IOP Publishing, p. 095007. ISSN: 0266-5611. DOI: 10.1088/1361-6420/ab1c09. URL: https://arxiv.org/abs/1901.05668 (visited on 04/22/2020).

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Number *Page*

LIST OF TABLES

Number *Page*

Chapter 1

INTRODUCTION

1.1 Background

Dynamical systems offer a powerful mathematical foundation for describing systems that evolve in time. Broadly, they can be defined as any class of recurrence relations (i.e., maps and flows), and common examples include systems of partial, ordinary, and stochastic differential equations. Such models date back to Newtonian Mechanics, and their early mathematical study can be traced to Poincaré. Dynamical systems have since been used to model temporal processes in nearly every discipline of science and engineering.

While mathematical analysis of dynamical systems brought clarity and predictive power to many fields, the computational revolution of the past 75 years—paired with an evolving suite of numerical algorithms—has allowed us to simulate impressively complex dynamics that were previously unattainable (e.g., global weather, epidemic spread, aircraft flight, human endocrine dynamics). Moreover, the increasing availability of relevant and timely data streams has provided tremendous opportunities for grounding such dynamical models in observations (i.e., via *data assimilation algorithms*), resulting in actionable forecasts (e.g., impending storms and viral spread) and automatic control systems (e.g., aircraft autopilot and automated insulin delivery). These successes crucially rely on: 1) sufficiently reliable dynamical models (i.e., an accurate mathematical description of relevant governing mechanisms) and 2) adequately informative data. From a Bayesian perspective, we can view (1) as our prior knowledge and (2) as our data—both are held in balance, and uncertainty in one can be counter-balanced by confidence in the other to yield a similarly accurate posterior estimate. When (1) and/or (2) is severely lacking, forecasting fidelity can drop to un-usable levels.

Over the past decade, however, we have witnessed a data-driven revolution in which *machine learning* methods can overcome a complete lack of (1) with an overwhelming abundance of (2) by approximating the unknown functions that generate the data. Indeed, substantial progress has been made towards learning dynamical systems from data, but these approaches tend to fail without exceptionally rich data sources.

Despite the growing availability of data streams, many processes that we strive to model, predict, and control have remained difficult to observe. Often times, quantities of interest are inaccessible or are captured by heavily biased measurements. *Biomedicine* provides countless examples of such challenges, where many physiologic processes are nearly impossible to measure (without causing harm) and the data that are collected (e.g., in a hospital's electronic health record database) can be deeply biased (i.e., missing not-at-random, recorded out of order, and designed to justify insurance billing). Thus, purely data-driven approaches often fail in biomedical settings.

Fortunately, it is possible to compensate for reduced data-fidelity with increased prior knowledge. In the context of dynamical systems, this often comes in the form of approximate mechanistic models, as well as physical properties of the model or its solution (e.g., conservation laws, boundedness, etc.). Thus, a key current challenge is to leverage and combine existing paradigms for model-based and data-driven inference to cope with common scenarios in which both our knowledge (1) and our data (2) are limited, yet complimentarily informative. Applications in biomedicine and geophysics, in particular, stand to benefit from such innovation. This thesis studies these scenarios from various angles, providing novel application-neutral approaches for hybridizing mechanisms and data, application-specific approaches for improved modeling of the glucose-insulin system, and new insights into computations that can be performed by biology itself:

- Chapter 2 focuses on infusing additional knowledge (specifically, state-based constraints) into a data assimilation algorithm (specifically, the Ensemble Kalman Filter) to create a contrained Ensemble Kalman Filter (cEnKF). This proves especially valuable when missing key observational components, creating substantial model un-identifiabilities. We find this in the case of observing patient blood glucose levels (we cannot measure their blood insulin levels) and assimilating this information into dynamical models of the glucose-insulin system. The numerical results show that incorporating simple state-based constraints via the proposed algorithm improves fidelity of glucose forecasting.
- Chapter 3 focuses specifically on the challenge of modeling patient-specific glucose dynamics with limited patient-level data. While Chapter 2 addresses the limited data problem by constraining the inference, Chapter 3 explicitly proposes a new data-driven model (derived from a Ornstein-Uhlenbeck stochastic

differential equation) that is identifiable under challenging data constraints and uses state-based stochasticity to capture the uncertainty induced by such limited data. The numerical results suggest non-inferiority of the proposed simple linear model when compared to state-of-the-art non-linear models (paired with non-linear data assimilators) in data-poor settings.

- Chapter 4 takes a broad view of hybrid modeling of dynamical systems in which we fuse incomplete models with data. First, it provides a substantial literature review on the topic. Then, it rigorously analyzes hybrid modeling under idealized data assumptions (i.e., fully-observed continuously in time without noise). Finally, it proposes and evaluates a novel algorithm for learning hybrid ordinary differential equations from noisy, partially-observed, irregularly spaced timeseries data by embedding auto-differentiable data assimilation algorithms within a neural ordinary differential equations inference framework.

- Chapter 5 identifies a common infidelity amongst mechanistic models of glucose-insulin dynamics and applies the methodology proposed in Chapter 4 to learn data-driven model corrections. Specifically, we recognize that even state-of-the-art mechanistic models of glucose-insulin dynamics fail to fully capture how the glucose-insulin system responds to different meal-types (e.g., a sandwich versus lasagna). Popular models are designed to take only carbohydrate quantities as inputs, but not other macronutrients (e.g., fat, protein, fiber); this is likely due to a lack of experimentally-identified mechanisms for such effects. However, these effects are empirically well-characterized (i.e., via glycemic index), and ought to be learnable from data. To test this hypothesis, we use the methodology proposed in Chapter 4 to learn targeted nutrition-specific corrections to the Bergman minimal model using simulated data.

- Chapter 6 reviews and unifies a variety of approaches for learning structural errors in models of dynamical systems. In contrast to Chapter 4, which focuses on learning deterministic models from trajectory-based data, this work takes a broader view. For example, we i) admit data that are either temporal or come from time-averaging, ii) consider deterministic and stochastic closure models, iii) discuss common choices for model parameterization, iv) discuss gradient-based and gradient-free optimization techniques for learning, and v) discuss how to incorporate physical laws and constraints into the learning. We have written this manuscript for accessibility to a broad geophysical readership.

- Chapter 7 focuses on viewing protein dimerization networks in biology through the mathematical lens of function approximation theory. In particular, we view these chemical reactions as functions that map initial concentrations of protein monomers to equilibrium concentrations of protein dimers, which are of interest due to their known biochemical activities. We use numerical simulations of these networks to examine the range of input-output functions that can be computed by different networks. These computational experiments allow us to characterize the *expressivity* and *versatility* of input-output maps induced by combinatorial protein dimerization networks, and we study these properties as a function of both network size and connectivity. Understanding these properties may allow for the prediction and control of natural cellular behaviors and enable the design of synthetic circuits.

1.2 Contributions

The chapters of this thesis are derived from manuscripts written for publication in journals or conference proceedings.

Chapter 2 is primarily derived from [1], and highlights its contributions in constrained Ensemble Kalman Filtering (cEnKF), and applications of this approach to forecasting problems in blood glucose dynamics. Thus, sections in [1] that discuss constrained Ensemble Kalman Inversion (cEKI) and its applications to geophysical problems has been omitted. Furthermore, Section 2.5.2 of Chapter 2 includes additional blood glucose forecasting results that are excerpted from [2]; these results come from direct application of the methodologies and models described in Chapter 2 and [1].

My contributions to [1] (published in Inverse Problems in 2019) were as follows:

- Writing of algorithms for cEnKF and cEKI in the paper
- Implementation of numerics for cEnKF and blood glucose model for real patient data
- Co-writing and co-editing the manuscript

My contributions to [2] (Journal of Biomedical Informatics in 2022) were as follows:

- Conceptualization and design of experiments
- Implementation of computational infrastructure for performing experiments

- Co-writing and co-editing the manuscript

Chapter 3 is taken directly from [6] (submitted to Chaos in 2023), and focuses on the introduction of a simple stochastic differential equation (SDE) model for blood glucose dynamics. The key insight of this work is the recognition that typical ODE models for blood glucose dynamics are so un-identifiable in some low-data regimes that it may, instead, be beneficial to work with an identifiable reduced-order model (our SDE). My contributions to [6] were as follows:

- Conceptualization of underpinning mathematical model
- Conceptualization of experiments, which compare to other state-of-the-art approaches
- Design and implementation of computational infrastructure for performing experiments
- Co-writing and co-editing the manuscript

Chapter 4 is taken directly from [5] (published in Communications of the American Mathematical Society in 2022), and focuses on learning hybrid (i.e., physics-based and data-driven) models of dynamical systems from partially-observed, noisy data. My contributions to [5] were as follows:

- Conceptualization of underpinning mathematical framework
- Invention of novel algorithms for learning dynamics from partially-observed and/or noisy data (i.e., leveraging autodifferentiable data assimilation)
- Writing and research of comprehensive review on the topic (this paper has substantial review content in the first sections)
- Conceptualization and design of numerical experiments
- Writing and editing the manuscript

Chapter 5 is taken directly from [7], and focuses on applying and evaluating the novel methods in Chapter 4 in an applied glucose-insulin modeling setting. My contributions to [5] were as follows:

- Conceptualization of underpinning mathematical model
- Conceptualization and design of numerical experiments
- Implementation of methods in [5] for the specific glucose-insulin model.
- Co-writing and co-editing the manuscript

Chapter 6 is taken directly from a manuscript that is currently in preparation [3]. This work reviews and unifies a variety of approaches for learning structural errors in models of dynamical systems, and is aimed towards a geophysical audience. My contributions to [3] are as follows:

- Conceptualization of underpinning mathematical framework
- Conceptualization and implementation of numerical experiments for learning structural error terms from non-ergodic systems using timeseries data. This extends concepts in Chapter 4 to derivative-free optimization settings.
- Co-writing and co-editing the manuscript

Chapter 7 describes ongoing joint work with Michael Elowitz, Pietro Perona, Andrew Stuart, Jacob Parres-Gold, and Benjamin Emert, for which a manuscript is currently being prepared [4]. This project focuses on viewing protein dimerization networks in biology through the mathematical lens of function approximation theory. In particular, we are interested in the expressivity and versatility of mathematical functions that can be realized through biological networks. My contributions to [4] are as follows:

- Conceptualization of underpinning mathematical framework
- Conceptualization and implementation of numerical experiments for evaluating quantitative metrics for network expressivity and versatility using simulations of protein dimerization reactions.
- Co-writing and co-editing the manuscript

1.3 Notation

Throughout the paper we use $\mathbb{N}$ to denote the positive integers $\{1, 2, 3, \cdots\}$ and $\mathbb{Z}^+$ to denote the non-negative integers $\mathbb{N} \cup \{0\} = \{0, 1, 2, 3, \cdots\}$. The matrix I_M denotes the identity on $\mathbb{R}^M$. We use $|\cdot|$ to denote the Euclidean norm, and the corresponding inner-product is denoted $\langle \cdot, \cdot \rangle$. A symmetric, square matrix A is positive definite (resp. positive semi-definite) if the quadratic form $\langle u, Au \rangle$ is positive (resp. non-negative) for all $u \neq 0$. By $|\cdot|_B$ we denote the weighted norm defined by $|v|_B^2 = v^* B^{-1} v$ for any positive-definite B. The corresponding weighted Euclidean inner-product is given by $\langle \cdot, \cdot \rangle_B := \langle \cdot, B^{-1} \cdot \rangle$. We use $\otimes$ to denote the outer product between two vectors: $(a \otimes b)c = \langle b, c \rangle a$.

References

[1] David J. Albers, Paul-Adrien Blancquart, Matthew E. Levine, Elnaz Esmaeilzadeh Seylabi, and Andrew Stuart. “Ensemble Kalman methods with constraints”. en. In: *Inverse Problems* 35.9 (Aug. 2019). Publisher: IOP Publishing, p. 095007. ISSN: 0266-5611. DOI: 10.1088/1361-6420/ab1c09. URL: https://arxiv.org/abs/1901.05668 (visited on 04/22/2020).

[2] David J. Albers, Melike Sirlanci, Matthew E. Levine, Jan Classen, Caroline Der Nigoghossian, and George Hripcsak. “Interpretable Forecasting of Physiology in the ICU Using Constrained Data Assimilation and Electronic Health Record Data”. In: *Journal of Biomedical Informatics (In review)* (2023). arXiv: 2305.06513 [stat.AP].

[3] D Huang, ME Levine, T Schneider, Z Shen, AM Stuart, and J Wu. “Learning About Structural Errors in Models of Complex Dynamical Systems”. In: *In preparation* (2023).

[4] J Parres-Gold, B Emert, ME Levine, P Perona, AM Stuart, and M Elowitz. “On the expressivity of combinatorial protein dimerization networks”. In: *In preparation* (2023).

[5] ME Levine and AM Stuart. “A Framework for Machine Learning of Model Error in Dynamical Systems”. In: *Communications of the American Mathematical Society* 2.07 (2022), pp. 283–344. ISSN: 2692-3688. DOI: 10.1090/cams/10.

[6] M. Sirlanci, M. E. Levine, C. C. Low Wang, D. J. Albers, and A. M. Stuart. “A Simple Modeling Framework For Prediction In The Human Glucose-Insulin System”. In: *Chaos (To appear)* (2023). arXiv: 1910.14193 [q-bio.QM].

[7] Ke Alexander Wang, Matthew E. Levine, Jiaxin Shi, and Emily B. Fox. “Learning Absorption Rates in Glucose-Insulin Dynamics from Meal Covariates”. In: *NeurIPS 2022 Workshop on Learning from Time Series for Health*. 2023. arXiv: 2304.14300 [cs.LG].

Chapter 2

ENSEMBLE KALMAN METHODS WITH CONSTRAINTS

2.1 Introduction

One of the key challenges in predicting physiologic dynamics is reconciling available data with pre-existing mechanistic models to provide greater insight and enhanced predictions. Data assimilation (DA) [225] offers a mathematical framework for striking this balance, and has been successfully leveraged in a variety of physiologic contexts; we introduce this framework in Section 2.2.

However, the performance of DA-based methods has been limited, in part, due to the underdetermined inference problem that results from projecting infrequent, noisy, partially-observed data streams onto large, highly parameterized physiologic models. We can address this identifiability challenge by either constraining the inference (the focus of Section 2.4) or simplifying the model (the focus of Chapter 3). In Section 2.4, we derive a novel online data assimilation method that incorporates convex constraints into its inference of physiologic states and parameters. We evaluate this method as a replacement for an existing data assimilation scheme developed for forecasting blood glucose levels in people with diabetes [10] using identical models and datasets as the original work.

Remark 2.1.1. This chapter is derived from the manuscript published by Albers, Blancquart, Levine, Seylabi, and Stuart [9], and contains both excerpts and additions to that work. The numerics section is supplemented by results from Albers, Sirlanci, Levine, Classen, Der Nigoghossian, and Hripcsak [11].

2.1.1 Overview

Kalman filter based methods have been enormously successful in both state and parameter estimation problems. However, a major disadvantage of such methods is that they do not naturally take constraints into account. The ability to constrain a system often has a number of advantages that can play an important role in state and parameter estimation: they can be used to enforce physicality of modeled systems (non-negativity of physical quantities, for example); relatedly they can be used to ensure that computational models are employed only within state and parameter regimes where the model is well-posed; and finally the application of constraints may provide robustness to outlier data. Resulting improvements in algorithmic efficiency and performance, by means of enforcing constraints, has been demonstrated in the recent literature in a diverse set of fields, including process control [405], biomechanics [45], cell energy metabolism [147], medical imaging [232], engine health estimation [387], weather forecasting [191], chemical engineering [444], and hydrology [422].

In the probabilistic view of filtering methods, constraints may be introduced by moving beyond the Gaussian assumptions that underpin Kalman methods and imposing constraints through the prior distributions on states and/or parameters. This, however, can create significant computational burden as the resulting distributions cannot be represented in closed form, through a finite number of parameters, in the way that Gaussian distributions can be. In this paper, we circumvent this issue by taking the viewpoint that ensemble Kalman methods constitute a form of derivative-free optimization methodology, eschewing the probabilistic interpretation. The ensemble is used to calculate surrogates for derivatives. With this optimization perspective, constraints may be included in a natural way. Standard ensemble Kalman methods employ a quadratic optimization problem encapsulating relative strengths of belief in the predictions of the model and the data; these optimization problems have explicit analytic solutions. To impose constraints the optimization problem is solved only within the constraint set; when the constraints form a non-empty closed convex set, this constrained optimization problem has a unique solution.

In this introductory section, we give a literature review describing existing work in this setting, we describe the contributions in this paper, and we outline notation used throughout.

2.1.2 Literature Review

Overviews of state estimation using Kalman based methods may be found in [126, 345, 225, 70]. The focus of this article is on ensemble based Kalman methods, introduced by Evensen in [127] and further developed in [65, 126]. The extension of the ensemble Kalman methodology to parameter estimation and inverse problems is overviewed in [302], especially for oil reservoir applications, and in an application-neutral formulation in [188]. Equipping Kalman-based methods with constraints can be desirable for a variety of inter-linked reasons described in the previous subsection: to enforce known physical boundaries in order to improve estimation accuracy; to operationalize filtering of a model which is ill-posed in subsets of its state or parameter space; and to provide robustness to noisy data and outlier events.

In extending the Kalman filter to non-Gaussian settings, a number of methods may be considered. Particle filters provide the natural methodology if propagation of probability distributions is required for state [112] or parameter [104] estimation. In the optimization setting, there are three primary methodologies: the extended Kalman filter, the unscented Kalman filter and the ensemble Kalman filter. The extended Kalman filter is based on linearization of the nonlinear system and therefore needs the computation of derivatives for propagation of the state covariance; this makes them unattractive in high dimensional problems. Unscented and ensemble Kalman filters, on the other hand, can be considered as particle-based methods which are derivative-free. In the unscented Kalman filter, the particles (sigma points) are chosen deterministically and are propagated through the nonlinear system to approximate the covariance, which is then corrected using the Kalman gain to compute the new sigma points. In the ensemble Kalman filter, the particles (ensemble members) are chosen randomly from the initial ensemble and are propagated through the dynamical system and corrected using the Kalman gain without needing to maintain the covariance.

In [386], and more recently in [15], overviews of different ways to impose constraints in linear and nonlinear state estimation are presented. To ensure that the estimates satisfy the constraints, moving horizon based estimators that solve a constrained optimization problem have been proposed [355, 341]. The paper [412] proposed a recursive nonlinear dynamic data reconciliation (RNDDR) approach based on extended Kalman filtering to ensure that state and parameter estimates satisfy the imposed bounds and constraints. The updated state estimates in this method are obtained by solving an optimization problem instead of using the Kalman gain. The resulting covariance calculations are, however, still similar to the Kalman filter: that is, unconstrained propagation and correction involving the Kalman gain, which can affect the accuracy of the estimates. To eliminate this deficiency, [241] proposed a Kullback-Leibler based method to update states and error covariances by solving a convex optimization problem involving conic constraints.

On the other hand, the paper [411] combined the concept of the unscented transformation [194] with the RNDDR formulation. In the prediction step, they propose step sizes to scale sigma points asymmetrically to better approximate the covariance information in the presence of lower and upper bounds. Then, for the update of each sigma point, they solve a constrained optimization problem. One disadvantage of this procedure is that the chosen step sizes for scaling the sigma points can only ensure the bound constraints. The paper [405] also tested various algorithms based on constrained optimization, projection [388] and truncation [387] to enforce bound constraints on unscented Kalman filtering. The paper [278] developed a class of estimators named constrained unscented recursive estimators to address the limitations of the unscented RNDDR method using optimization-based projection algorithms for obtaining sigma points in the presence of convex, non-convex and bound constraints.

As mentioned earlier, since the corrected covariance is used to compute the sigma points, unscented formulations always require enforcing constraints in both propagation and correction/update steps. In contrast, ensemble-based methods only require constraints to be enforced in the update step. In this context, the paper [422] tested projection and accept/reject methods to constrain ensemble members in a post-processing step, after application of the unconstrained ensemble Kalman filter. In the former, they project the updated ensemble members to the feasible space if they violate the constraints and in the latter they enforce the updated ensemble members to obey the constraints by resampling the dynamic and/or data model errors. On the other hand, [327, 326] proposed updating the state estimates in ensemble Kalman filtering by solving a constrained optimization problem while truncating the Gaussian distribution of the initial ensemble. The paper [191] demonstrated how to enforce a physics-based conservation law on an ensemble Kalman filtering based state estimation problem by formulating the filter update as a set of quadratic programming problems arising from a linear data acquisition model subject to linear constraints. Here we develop this body of work on constraining ensemble Kalman techniques, providing a unifying framework with an underpinning theoretical basis.

2.1.3 Our Contribution

The preceding literature review demonstrates that the imposition of constraints on state and parameter estimation procedures is highly desirable. It also indicates that ensemble Kalman methods offer the most natural context in which to attempt to do this, as extended Kalman methods do not scale well to high dimensional state or parameter space, whilst the unscented filter does not lend itself as naturally to the incorporation of constraints.

In this paper we build on the application-specific papers [422, 191] which demonstrate how to impose some specific constraints on ensemble based parameter and state estimation problems respectively. We formulate a very general methodology which is application-neutral and widely applicable, thereby making the ideas in [422, 191] accessible to a wide community of researchers working in inverse problems and state estimation. We also describe a straightforward mathematical analysis which demonstrates that the resulting algorithms are well-defined since they involve the solution of quadratic minimization problems subject to convex constraints at each step of the algorithm; these optimization problems have a unique solution. And finally we showcase the methodology on two applications, one from biomedicine and one from seismology.

Sections 2.2 to 2.4 outlines the ensemble Kalman (EnKF) methodology for state estimation, with and without constraints. Section 2.5 describes the numerical experiments which illustrate the foregoing ideas.

2.2 Data Assimilation

Here we briefly introduce the framework for data assimilation given by Law, Stuart, and Zygalakis [225], which considers the dynamics, Ψ, of an underlying process v, which is noisily measured by a linear operator H to acquire measurements y:

$$\begin{aligned}
&\text{Dynamics Model:} && v_{j+1} = \Psi(v_j) + \xi_j, \quad j \in \mathbb{Z}^+ \\
&\text{Data Model:} && y_{j+1} = Hv_{j+1} + \eta_{j+1}, \quad j \in \mathbb{Z}^+ \\
&\text{Probabilistic Structure:} && v_0 \sim \mathcal{N}(m_0, C_0), \quad \xi_j \sim \mathcal{N}(0, \Sigma), \quad \eta_j \sim \mathcal{N}(0, \Gamma) \\
& && v_0 \perp \{\xi_j\} \perp \{\eta_j\} \text{ independent.}
\end{aligned} \tag{2.1}$$

We assume that $\mathcal{H}_1, \mathcal{H}_2$ are separable Hilbert spaces. Then $v_j \in \mathcal{H}_1$, and $\Psi : \mathcal{H}_1 \mapsto \mathcal{H}_1$ is the state-transition operator. The operator $H : \mathcal{H}_1 \mapsto \mathcal{H}_2$ is the linear observation operator and $y_j \in \mathcal{H}_2$. The covariance operators C_0, Σ are assumed trace-class on $\mathcal{H}_1$, and Γ on $\mathcal{H}_2$ which ensures that the initial condition v_1 and the noises ξ_j and η_j live in $\mathcal{H}_1, \mathcal{H}_1$ and $\mathcal{H}_2$ (respectively) with probability one.

Here $\xi_j \sim \mathcal{N}(0, \Sigma)$, $\eta_j \sim \mathcal{N}(0, \Gamma)$ are assumed independent gaussian noises in the model state dynamics and measurement operations with mean zero and covariances Σ and Γ, respectively. We also assume a known distribution $\mathcal{N}(m_0, C_0)$ for initial condition v_0.

Remark 2.2.1. Throughout this chapter we derive our theoretical results in the setting where $\mathcal{H}_1$ and $\mathcal{H}_2$ are finite dimensional; however the update formulae we derive are well-defined in the general Hilbert space setting and this fact is important because it means that the methods derived have a robustness to mesh refinement and similar procedures arising when the problem of interest is specified via a partial differential equation, or other infinite dimensional problem.

Remark 2.2.2. We restrict attention to linear observation operators H because this leads to solvable quadratic optimization problems within the context of Kalman-based methods. In principle, a non-linear observation operator could be used, but the optimization problems defining the algorithms arising in this work might not have a unique solution in this setting.

The problem of *filtering* aims to use historical measurements $Y_k := \{y_j\}_{j=1}^k$ to estimate the current state v_k—that is, to estimate the probability of $v_k|Y_k$. The *forecasting* problem is highly related, and aims to estimate $v_{k+1}|Y_k$; it typically does this by solving the filtering problem, then iterating the dynamics forward over the filtered distribution.

Filtering and prediction can be performed with many methods, which are surveyed in [225]. In settings where the dynamics Ψ are linear, the classical Kalman Filter [201] and its descendants are often the best choice. Indeed, these methods also lend themselves well to constrained state estimation [386]. Systems with non-linear Ψ, on the other hand, are much harder to filter, and have given rise to a wide array of non-linear stochastic filters. One of the most popular, which we focus on, is the Ensemble Kalman Filter (EnKF) [128], which can be highly advantageous for its derivative-free nature. Similar algoriths, such as the Unscented Kalman Filter (UKF) [194] are also worth studying; for example, the UKF was successfully deployed for glucose prediction problems in people with type 2 diabetes in [10]. Here, we focus on Ensemble Kalman Filtering, and introduce a rigorous approach to constraining its inference, as presented by Albers et al. [11]. We note that [11] also presents an analogous constraint methodology for Ensemble Kalman Inversion techniques, which we do not discuss here.

2.3 Ensemble Kalman Filter

The ensemble Kalman filter is a particle-based sequential optimization approach to the state estimation problem. The particles are denoted by $\{v_j^{(n)}\}_{n=1}^N$ and represent a collection of N candidate state estimates at time j. The method proceeds as follows. The state of all the particles at time $j+1$ are predicted using the dynamics model to give $\{\widehat{v}_{j+1}^{(n)}\}_{n=1}^N$. The resulting empirical covariance of the particles is then used to define the objective function $I_{\text{filter},j,n}(v)$, which encapsulates the model-data compromise. This is minimized in order to obtain the updates $\{v_{j+1}^{(n)}\}_{n=1}^N$.

The prediction step is

$$\widehat{v}_{j+1}^{(n)} = \Psi(v_j^{(n)}) + \xi_j^{(n)}, n = 1, ..., N \tag{2.2a}$$

$$\widehat{m}_{j+1} = \frac{1}{N}\sum_{n=1}^{N}\widehat{v}_{j+1}^{(n)} \tag{2.2b}$$

$$\widehat{C}_{j+1} = \frac{1}{N}\sum_{n=1}^{N}\big(\widehat{v}_{j+1}^{(n)} - \widehat{m}_{j+1}\big)\big(\widehat{v}_{j+1}^{(n)} - \widehat{m}_{j+1}\big)^T. \tag{2.2c}$$

Here we have $\xi_j^{(n)} \sim \mathcal{N}(0, \Sigma)$ i.i.d.. Because the empirical covariance contains only $N-1$ independent pieces of information, (2.2c) is sometimes scaled by $N-1$ and not N; making this change would lead to no changes in the statements and proofs of all the theorems, and would only affect the definition of covariance within the algorithms.

Let $\mathcal{R}(\widehat{C}_{j+1})$ denote the range of $\widehat{C}_{j+1}$. The update step is then

$$v_{j+1}^{(n)} = \underset{v}{\operatorname{argmin}}\, I_{\text{filter},j,n}(v) \tag{2.3}$$

where

$$I_{\text{filter},j,n}(v) := \begin{cases} \frac{1}{2} \mid y_{j+1}^{(n)} - Hv \mid_\Gamma^2 + \frac{1}{2} \mid v - \widehat{v}_{j+1}^{(n)} \mid_{\widehat{C}_{j+1}}^2 & \text{if } v - \widehat{v}_{j+1}^{(n)} \in \mathcal{R}(\widehat{C}_{j+1}). \\ \infty & \text{otherwise.} \end{cases} \tag{2.4}$$

It can be useful to rewrite the objective function for the optimization problem in an equivalent and more standard form for input to software:

$$\begin{cases} \frac{1}{2}v^T\Big(H^T\Gamma^{-1}H + \widehat{C}_{j+1}^{-1}\Big)v - \Big(\widehat{C}_{j+1}^{-1^T}\widehat{v}_{j+1}^{(n)} + H^T\Gamma^{-1^T}y_{j+1}^{(n)}\Big)^T v & \text{if } v - \widehat{v}_{j+1}^{(n)} \in \mathcal{R}(\widehat{C}_{j+1}). \\ \infty & \text{otherwise.} \end{cases}$$

The $y_{j+1}^{(n)}$ are either identical to the data y_{j+1}, or found by perturbing it randomly.

Note that $\widehat{C}_{j+1}$ is an operator of rank at most $N-1$, and thus can only be invertible when $N-1$ is larger than the dimension of $\mathcal{H}_1$. For moderate- and high-dimensional systems, it is often impractical to satisfy this condition. However, the minimizing solution can be found by regularizing $\widehat{C}_{j+1}$ by addition of ϵI for $\epsilon > 0$, deriving the update equations and then letting $\epsilon \to 0$. We give the resulting formulae, and then justify them immediately afterwards, in the following subsubsection. Alternatively it is possible to directly seek a solution in $\mathcal{R}(\widehat{C}_{j+1})$, which is a subspace of dimension $N-1$; this is done in the subsequent subsubsection.

Formulation In The Original Variables

The well-known Kalman update formulae arising from solution of the minimization problem (2.4) are as follows:

$$S_{j+1} = H\widehat{C}_{j+1}H^T + \Gamma \tag{2.5a}$$

$$K_{j+1} = \widehat{C}_{j+1}H^T S_{j+1}^{-1} \qquad (\text{Kalman G}ain) \tag{2.5b}$$

$$y_{j+1}^{(n)} = y_{j+1} + s\eta_{j+1}^{(n)}, n = 1, ..., N \tag{2.5c}$$

$$v_{j+1}^{(n)} = (I - K_{j+1}H)\widehat{v}_{j+1}^{(n)} + K_{j+1}y_{j+1}^{(n)}, n = 1, ..., N \tag{2.5d}$$

Here $\eta_j^{(n)} \sim \mathcal{N}(0, \Gamma)$ i.i.d. and the constant s takes value 0 or 1. When $s = 1$ the $y_{j+1}^{(n)}$ are referred to as perturbed observations. The choice $s = 1$ is made to ensure the correct statistics of the updates in the linear Gaussian setting when a probabilistic viewpoint is taken, and more generally to introduce diversity into the ensemble procedure when an optimization viewpoint is taken. Derivation of the formulae may be found in [225]. In brief the formulae arise from completing the square in the objective function $I_{\text{filter},j,n}(\cdot)$ and then applying the Sherman–Morrison formula to rewrite the updates in the data space rather than state space; the latter is advantageous in many applications where $\mathcal{H}_2$ has dimension much smaller than $\mathcal{H}_1$.

We summarize with the following pseudo-code:

Algorithm 1 EnKF Algorithm

```
1: Choose $\{v_0^{(n)}\}_{n=1}^N$, $j = 0$
2: Predict $\{\widehat{v}_{j+1}^{(n)}\}_{n=1}^N$, $\widehat{C}_{j+1}$ from (2.2)
3: Update $\{v_{j+1}^{(n)}\}_{n=1}^N$ from (2.5)
4: $j \leftarrow j + 1$, go to 2.
```

An equivalent formulation of the minimization problem is now given by means of a penalized Lagrangian approach to incorporate the property that the solution of the optimization problem lies in the range of the empirical covariance. The perspective is particularly useful when further constraints are imposed on the solution of the optimization problem.

Theorem 2.3.1. *Suppose that the dimensions of $\mathcal{H}_1$ and $\mathcal{H}_2$ are finite. Let j be in $\mathbb{Z}^+$ and $1 \leq n \leq N$. Define $y' = y_{j+1}^{(n)} - H\widehat{v}_{j+1}^{(n)}$. Then the update formulae* (2.2), (2.5) *may be given alternatively by*

$$v_{j+1}^{(n)} = \widehat{v}_{j+1}^{(n)} + \underset{(a,v')\in\mathcal{A}}{\operatorname{argmin}} \left(\frac{1}{2} \mid y' - Hv' \mid_{\Gamma}^{2} + \frac{1}{2}\langle a, v' \rangle \right) \tag{2.6}$$

where $\mathcal{A} = \{(a, v') \in \mathcal{H}_1 \times \mathcal{H}_1 : \widehat{C}_{j+1} a = v'\}$ and the argmin is projected from the pair (a, v') onto the v' coordinate only. Moreover $v_{j+1}^{(n)} = \lim_{\epsilon \to 0} v_\epsilon$ with

$$v_\epsilon = \underset{v\in\mathcal{H}_1}{\operatorname{argmin}} \left(\frac{1}{2} \mid y_{j+1}^{(n)} - Hv \mid_{\Gamma}^{2} + \frac{1}{2} \mid \widehat{v}_{j+1}^{(n)} - v \mid_{\widehat{C}_\epsilon}^{2} \right)$$

and $\hat{C}_\epsilon = \widehat{C}_{j+1} + \epsilon I$.

Proof. For notational convenience denote $\widehat{C} = \hat{C}_{j+1}$ and see that the minimization (2.6) is performed under the constraint $\widehat{C}a = v'$. Then notice that $\langle a, v' \rangle =\mid v' \mid_{\widehat{C}}^{2}$ with v' lying in the range of the operator $\widehat{C}$; this is a convex constraint. The restriction of $\widehat{C}$ over the constraint set is positive definite which means that the quadratic objective function, now depending only on v', is strongly convex. Therefore the problem has a unique solution and its Lagrangian is written as:

$$\mathcal{L}(v', a, \lambda) = \frac{1}{2}|y' - Hv'|_{\Gamma}^{2} + \frac{1}{2}\langle a, v' \rangle + \langle \lambda, \widehat{C}a - v' \rangle$$

To express optimality conditions compute the derivatives and set them to zero:

$$\begin{aligned} -H^T\Gamma^{-1}(y' - Hv') + \frac{1}{2}a - \lambda &= 0, \\ \frac{1}{2}v' + \widehat{C}\lambda &= 0, \\ v' - \widehat{C}a &= 0. \end{aligned}$$

The last two equations imply that $\widehat{C}(2\lambda + a) = 0$. Thus we set $\lambda = -\frac{1}{2}a$ and drop the second equation, replacing the first by

$$-H^T\Gamma^{-1}(y' - H\widehat{C}a) + a = 0.$$

Solving this for a gives

$$\begin{aligned} v_{j+1}^{(n)} &= \hat{v}_{j+1}^{(n)} + v' \\ &= \hat{v}_{j+1}^{(n)} + \widehat{C}a \\ &= \hat{v}_{j+1}^{(n)} + \widehat{C}(H^T\Gamma^{-1}H\widehat{C} + I)^{-1}H^T\Gamma^{-1}y' \\ &= \hat{v}_{j+1}^{(n)} + \widehat{C}(H^T\Gamma^{-1}H\widehat{C} + I)^{-1}H^T\Gamma^{-1}(y_{j+1}^{(n)} - H\hat{v}_{j+1}^{(n)}) \\ &= (I - K_{j+1}H)\hat{v}_{j+1}^{(n)} + K_{j+1}y_{j+1}^{(n)}\,. \end{aligned}$$

It remains to show that K_{j+1} agrees with the prescription given in the formulae above. To see this we note that if we choose S to be any matrix satisfying $K_{j+1} = \widehat{C}H^T S^{-1}$ then

$$H^T S^{-1} = (H^T\Gamma^{-1}H\widehat{C} + I)^{-1}H^T\Gamma^{-1}$$

so that

$$(H^T\Gamma^{-1}H\widehat{C} + I)H^T = H^T\Gamma^{-1}S.$$

Thus

$$H^T\Gamma^{-1}H\widehat{C}H^T + H^T = H^T\Gamma^{-1}S$$

which may be achieved by choosing any S so that

$$\Gamma^{-1}(H\widehat{C}H^T + \Gamma) = \Gamma^{-1}S$$

and multiplication by Γ gives the desired formula for S_{j+1}.

Concerning the alternative representation of the solution, we note that $H^T\Gamma^{-1}H + \widehat{C}_\epsilon^{-1}$ is strictly positive definite and hence the related quadratic function is strongly convex. As a consequence we have existence and uniqueness of the solution, and the optimality condition becomes,

$$(H^T\Gamma^{-1}H + \widehat{C}_\epsilon^{-1})v_\epsilon = H^T\Gamma^{-1}y_{j+1}^{(n)} + \widehat{C}_\epsilon^{-1}\widehat{v}_{j+1}^{(n)}.$$

Then if we apply Woodbury matrix identity we obtain

$$v_\epsilon = (\widehat{C}_\epsilon - \widehat{C}_\epsilon H^T(H\widehat{C}_\epsilon H^T + \Gamma)^{-1}H\widehat{C}_\epsilon)(H^T\Gamma^{-1}y_{j+1}^{(n)} + \widehat{C}_\epsilon^{-1}\widehat{v}_{j+1}^{(n)})$$

and rearranging the terms:

$$v_\epsilon = (I - \hat{C}_\epsilon H^T(H\hat{C}_\epsilon H^T + \Gamma)^{-1}H)\hat{v}_{j+1}^{(n)} + \hat{C}_\epsilon H^T(H\hat{C}_\epsilon H^T + \Gamma)^{-1}y_{j+1}^{(n)}.$$

Finally, as $A \mapsto A^{-1}$ is continuous over the set of invertible matrices, letting $\epsilon \longrightarrow 0$ gives:

$$\lim_{\epsilon\longrightarrow 0} v_\epsilon = (I - K_{j+1}H)\hat{v}_{j+1}^{(n)} + K_{j+1}y_{j+1}^{(n)}$$

which concludes the proof. □

Formulation In Range Of The Covariance

The form of the minimization problem for each individual particle has a special structure which follows from the fact that the predicted covariance is computed empirically and is a sum of rank one matrices. This allows us to seek the solution of the minimization problem as a linear combination of a given set of vectors, and to minimize over the scalars which define this linear combination. This reformulation of the optimization problem is useful if the number of ensemble members N is much smaller than the dimension of the data space, where the inversion of S takes place to form Kalman gain K.

In order to implement the minimization in the N dimensional subspace we note that $I_{\text{filter},j,n}(v)$ is infinite unless

$$v - \widehat{v}_{j+1}^{(n)} = \widehat{C}_{j+1} a$$

for some $a \in \mathbb{R}^n$. From the structure of $\widehat{C}_{j+1}$ given in (2.2c) it follows that

$$v = \widehat{v}_{j+1}^{(n)} + \frac{1}{N} \sum_{m=1}^{N} b_m e^{(m)}, \quad e^{(m)} := \widehat{v}_{j+1}^{(m)} - \widehat{m}_{j+1}. \tag{2.8}$$

Here each unknown parameter $b_m \in \mathbb{R}$ and $b := \{b_m\}_{m=1}^{N}$, is the unknown vector to be determined. This form for v follows from the fact that

$$\widehat{C}_{j+1} = \frac{1}{N} \sum_{m=1}^{N} e^{(m)} \otimes e^{(m)} \tag{2.9}$$

which in turn implies that

$$\widehat{C}_{j+1} a = \frac{1}{N} \sum_{m=1}^{N} b_m e^{(m)}. \tag{2.10}$$

Note that the unknown vector b depends on n as we need to solve the constrained minimization problem for each of the particles, indexed by $n = 1, \ldots, N$; we have suppressed the dependence of b on n for notational simplicity.

The expression (2.8) for v in terms of the $e^{(m)}$ can be substituted into (2.4) to obtain a functional $J_{\text{filter},j,n}(b)$ to be minimized over $b \in \mathbb{R}^N$, because v is an affine function of b. Equation (2.8) may be written in compact form as

$$v = \widehat{v}_{j+1}^{(n)} + Bb \tag{2.11}$$

where B is the linear mapping from $\mathbb{R}^N$ into $\mathcal{H}_1$ defined by

$$Bb := \frac{1}{N} \sum_{m=1}^{N} b_m e^{(m)}.$$

We now identify $J_{\text{filter},j,n}(b)$. We note that (2.10) is solved by taking

$$b_m = \langle e^{(m)}, a \rangle.$$

Now note that

$$\frac{1}{2} \mid v - \widehat{v}^{(n)}_{j+1} \mid^2_{\widehat{C}_{j+1}} = \frac{1}{2}\langle a, \widehat{C}_{j+1} a\rangle = \frac{1}{2N}\sum_{m=1}^{N} b_m^2.$$

Using this and (2.11) in the definition of $I_{\text{filter},j,n}(v)$ we obtain

$$J_{\text{filter},j,n}(b) = I_{\text{filter},j,n}\big(\widehat{v}^{(n)}_{j+1} + Bb\big)$$

and hence, from (2.4),

$$J_{\text{filter},j,n}(b) := \frac{1}{2} \mid y^{(n)}_{j+1} - H\widehat{v}^{(n)}_{j+1} - HBb \mid^2_{\Gamma} + \frac{1}{2N}|b|^2 \tag{2.12a}$$

$$= \frac{1}{2} b^T \left(B^T H^T \Gamma^{-1} H B + \frac{1}{N} I\right) b - \left(B^T H^T \Gamma^{-1} (y^{(n)}_{j+1} - H\hat{v}^{(n)}_{j+1})\right)^T b + \text{const.} \tag{2.12b}$$

Once b is determined it may be substituted back into (2.11) to obtain the solution to the minimization problem.

The preceding considerations also yield the following result, concerning the unconstrained Kalman minimization problem; its proof is a corollary of the more general Theorem 2.4.1 from the next subsection, which includes constraints in the minimization problem.

Corollary 2.3.2. *Suppose that the dimensions of $\mathcal{H}_1$ and $\mathcal{H}_2$ are finite. Given the prediction (2.2a), the unconstrained Kalman update formulae may be found by minimizing $J_{\text{filter},j,n}(b)$ from* (2.12) *with respect to b and substituting into* (2.11).

We summarize the ensemble Kalman state estimation algorithm, using minimization over the vector b, in the following pseudo-code:

Algorithm 2 EnKF Algorithm formulated in range of covariance

```
1: Choose $\{v^{(n)}_0\}_{n=1}^N$, $j = 0$
2: Predict $\{\widehat{v}^{(n)}_{j+1}, e^{(n)}\}_{n=1}^N$, from (2.2)
3: Optimize $\{b^{(n)}\}_{n=1}^N$ as argmin of (2.12)
4: Update $v^{(n)}_{j+1} = \widehat{v}^{(n)}_{j+1} + Bb^{(n)}$ from (2.11)
5: $j \leftarrow j + 1$, go to 2.
```

2.4 Constrained Ensemble Kalman Filter

In this subsection we introduce linear equality and inequality constraints on the state variable into the ensemble Kalman filter. We make prediction according to (2.2), and then incorporate data by solving the minimization problem (2.4) subject to the additional constraints

$$Fv = f, \tag{2.13a}$$

$$Gv \preceq g. \tag{2.13b}$$

Here F and G are linear mappings which, respectively, take the state v into the number of equality and inequality constraints; the notation $\preceq$ denotes inequality componentwise.

Formulation In The Original Variables

The preceding considerations lead to the following algorithm for ensemble Kalman filtering subject to constraints (the theoretical justification for using this algorithm follows from Theorem 2.4.1 below):

Algorithm 3 Constrained EnKF Algorithm

```
1: Choose $\{v_0^{(n)}\}_{n=1}^N$, $j = 0$
2: Predict $\{\widehat{v}_{j+1}^{(n)}\}_{n=1}^N$, $\widehat{C}_{j+1}$ from (2.2)
3: Update $\{v_{j+1}^{(n)}\}_{n=1}^N$ from (2.5)
4: for $n = 1 : N$
5:     if $v_{j+1}^{(n)}$ violates constraints in (2.13)
6:         $v_{j+1}^{(n)} \leftarrow$ argmin of (2.4) subject to (2.13)
7:     end if
8: end for
9: $j \leftarrow j + 1$, go to 2.
```

Formulation In Range Of The Covariance

The linear constraints (2.13) can be rewritten in terms of the vector b, by means of (2.11), as follows:

$$FBb = f - F\widehat{v}_{j+1}^{(n)}, \tag{2.14a}$$

$$GBb \preceq g - G\widehat{v}_{j+1}^{(n)}. \tag{2.14b}$$

We may thus predict and then optimize the objective function $J_{\text{filter},j,n}(b)$, given by (2.12), subject to the constraints (2.14). Implementation of this leads to following algorithm for ensemble Kalman filtering subject to constraints:

Algorithm 4 Constrained EnKF Algorithm formulated in range of covariance

```
1: Choose $\{v_0^{(n)}\}_{n=1}^N$, $j = 0$
2: Predict $\{\widehat{v}_{j+1}^{(n)}, e^{(n)}\}_{n=1}^N$, from (2.2)
3: Update $b^{(n)} \leftarrow$ argmin of (2.12), $v_{j+1}^{(n)} = \widehat{v}_{j+1}^{(n)} + Bb^{(n)}$ from (2.11)
4: for $n = 1 : N$
5:     if $v_{j+1}^{(n)}$ violates constraints in (2.13)
6:         $b^{(n)} \leftarrow$ argmin of (2.12) subject to (2.14)
7:         Update $v_{j+1}^{(n)} = \widehat{v}_{j+1}^{(n)} + Bb^{(n)}$ from (2.11)
8:     end if
9: end for
10: $j \leftarrow j + 1$, go to 2.
```

Justification for the use of this algorithm, working in the constrained space parameterized by b, is a consequence of the following:

Theorem 2.4.1. *Suppose that the dimensions of $\mathcal{H}_1$ and $\mathcal{H}_2$ are finite. The problem of finding $v_{j+1}^{(n)}$ as the minimizer of $I_{\text{filter},j,n}(v)$ subject to the constraints (2.13) is equivalent to finding b to minimize $J_{\text{filter},j,n}(b)$ subject to the constraints (2.14) and then using (2.11) to find $v_{j+1}^{(n)}$ from b. Furthermore, both of these constrained minimization problems have a unique solution provided that the constraint sets are non-empty.*

Proof. For notational convenience set $\widehat{v} = \widehat{v}_{j+1}^{(n)}$, $y = y_{j+1}^{(n)}$, $y' = y - H\widehat{v}$, $\widehat{C} = \widehat{C}_{j+1}$ and $\widehat{C}_\epsilon = \widehat{C}_{j+1} + \epsilon I$.

Denote

$$\begin{aligned}
v^* = &\underset{v'}{\text{argmin}} && \frac{1}{2} \mid y' - Hv' \mid_\Gamma^2 + \frac{1}{2}\langle a, v' \rangle \\
&\text{subject to} && \bullet\, \widehat{C}a = v' \\
& && \bullet\, Fv' = f - F\widehat{v} \\
& && \bullet\, Gv' \leq g - G\widehat{v}
\end{aligned} \tag{2.15}$$

$$\begin{aligned}
v_\epsilon = &\underset{v}{\text{argmin}} && \frac{1}{2} \mid y - Hv \mid_\Gamma^2 + \frac{1}{2} \mid v - \widehat{v} \mid_{\widehat{C}_\epsilon}^2 \\
&\text{subject to} && \bullet\, Fv = f \\
& && \bullet\, Gv \leq g
\end{aligned} \tag{2.16}$$

and

$$\begin{aligned}
J(v) &= \frac{1}{2} \mid y - Hv \mid_\Gamma^2 + \frac{1}{2} \mid v - \widehat{v} \mid_{\widehat{C}}^2 \\
J_\epsilon(v) &= \frac{1}{2} \mid y - Hv \mid_\Gamma^2 + \frac{1}{2} \mid v - \widehat{v} \mid_{\widehat{C}_\epsilon}^2
\end{aligned}$$

The part of the statement of Theorem 2.4.1 concerning existence of a minimizer is a consequence of the Lemma 2.4.2 stated and proved below. The second part, concerning the equivalence of minimization over b and over v (or v') was shown prior to the theorem statement. This concludes the proof. □

Lemma 2.4.2. *Suppose that the constraint sets of* (2.15) *and* (2.16) *are non empty, then* v^* *exists and is unique and for all* $\epsilon > 0$, v_ϵ *exists and is unique. Furthermore* $\underset{\epsilon \to 0}{lim}\, v_\epsilon = v^* + \widehat{v}$.

Proof. To prove existence and uniqueness of the solution of (2.15), notice that it can be reformulated as

$$\begin{aligned}
&\underset{v'}{\text{argmin}} && J(v' + \widehat{v}) \\
&\text{subject to} && \bullet\, \widehat{C}a = v' \\
& && \bullet\, Fv' = f - F\widehat{v} \\
& && \bullet\, Gv' \leq g - G\widehat{v}
\end{aligned}$$

and that the restriction of $\widehat{C}$ over its range is strictly positive definite. Hence J is a strongly convex function being minimized over a non empty closed convex set. From standard theory v^* exists and is unique. Then as $\widehat{C}_\epsilon$ is strictly positive definite, the same type of arguments provide existence and uniqueness of v_ϵ.

Now we prove the second part of the lemma. We note that $v^* + \widehat{v}$ matches the constraints of (2.16). It follows that for all $\epsilon > 0$, $J_\epsilon(v_\epsilon) \leq J_\epsilon(v^* + \widehat{v})$. Then let us prove that $J_\epsilon(v^* + \widehat{v}) \underset{\epsilon \to 0}{\longrightarrow} J(v^* + \widehat{v})$. First denote by $\lambda_1 \leq \cdots \leq \lambda_{N-1}$ the strictly positive eigenvalues of $\widehat{C}$ (recall that $\widehat{C}$ is symmetric positive semidefinite and that $\text{rank}(\widehat{C}) = N - 1$ almost surely). Hence $\widehat{C}_\epsilon^{-1} = \sum_{k=1}^{N-1} \frac{1}{\lambda_k + \epsilon} a_k a_k^T + \sum_{k=N}^{\dim(\mathcal{H}_1)} \frac{1}{\epsilon} a_k a_k^T$ where the a_k's are the eigenvectors of $\widehat{C}$ (the first and second sums respectively gather the vectors of the range and of the nullspace of $\widehat{C}$) . As v^* lies in the range of $\widehat{C}$, it holds that $| \, v^* + \widehat{v} - \widehat{v} \, |^2_{\widehat{C}_\epsilon} = | \, v^* \, |^2_{\widehat{C}_\epsilon} = \sum_{k=1}^{N-1} \frac{1}{\lambda_k + \epsilon}(a_k^T v^*)^2$. Now as the a_k's do not depend on ϵ, by letting ϵ tending to zero, this quantity will tend to

$$\sum_{k=1}^{N-1} \frac{1}{\lambda_k}(a_k^T v^*)^2 = | \, v^* \, |^2_{\widehat{C}} = | \, v^* + \widehat{v} - \widehat{v} \, |^2_{\widehat{C}} \, .$$

Therefore it holds that $J_\epsilon(v^* + \widehat{v}) \underset{\epsilon \to 0}{\longrightarrow} J(v^* + \widehat{v})$. From this we deduce that there exists $\delta > 0$ such that for all $0 < \epsilon < \delta$, $J_\epsilon(v_\epsilon) \leq J(v^* + \widehat{v}) + 1$.

Then set $w_\epsilon = v_\epsilon - \widehat{v} = w_\epsilon^0 + w_\epsilon^1$ where w_ϵ^0 lies in the nullspace of $\widehat{C}$ and w_ϵ^1 in its range (recall that for a symmetric matrix nullspace and range are orthogonal) and see that $J_\epsilon(v_\epsilon) = \frac{1}{2} \, | \, y' - H w_\epsilon \, |^2_\Gamma + \frac{1}{2} \, | \, w_\epsilon \, |^2_{\widehat{C}_\epsilon}$. It holds that $\frac{1}{2} \, | \, w_\epsilon \, |^2_{\widehat{C}_\epsilon} \leq J_\epsilon(v_\epsilon) \leq J(v^* + \widehat{v}) + 1$ for ϵ sufficiently small. Furthermore $| \, w_\epsilon \, |^2_{\widehat{C}_\epsilon} = | \, w_\epsilon^0 \, |^2_{\widehat{C}_\epsilon} + | \, w_\epsilon^1 \, |^2_{\widehat{C}_\epsilon} = \frac{1}{\epsilon} \, | \, w_\epsilon^0 \, |^2 + | \, w_\epsilon^1 \, |^2_{\widehat{C}_\epsilon}$, and since this quantity is bounded from above we deduce that $w_\epsilon^0 \underset{\epsilon \to 0}{\longrightarrow} 0$ and that w_ϵ^1 is bounded. Let $(\epsilon_m)_{m \in \mathbb{N}}$ be a sequence of positive real numbers such that $\epsilon_m \underset{m \to \infty}{\longrightarrow} 0$, and from the preceding extract a converging subsequence (denoted $(\epsilon_m)_{m \in \mathbb{N}}$ for simplicity) such that $(w_{\epsilon_m}^1)_{m \in \mathbb{N}}$ converges to a limit denoted w^*. As $w_{\epsilon_m}^1$ lies in $\mathcal{R}(\widehat{C})$, we can use the eigenvalue decomposition of $\widehat{C}$ to show that $| \, w_{\epsilon_m}^1 \, |^2_{\widehat{C}_{\epsilon_m}} \underset{m \to \infty}{\longrightarrow} | \, w^* \, |^2_{\widehat{C}}$. This limiting identity, and the fact that w_ϵ^0 has limit 0, may be used to establish the first equality within the following chain of equalities and inequalities:

$$\begin{aligned} J(w^* + \widehat{v}) &= \lim_{m \to \infty} \frac{1}{2} \, | \, y' - H w_{\epsilon_m} \, |^2_\Gamma + \frac{1}{2} \, | \, w_{\epsilon_m}^1 \, |^2_{\widehat{C}_{\epsilon_m}} \leq \lim_{m \to \infty} J_{\epsilon_m}(v_{\epsilon_m}) \\ &\leq \lim_{m \to \infty} J_{\epsilon_m}(v^* + \widehat{v}) = J(v^* + \widehat{v}). \end{aligned}$$

Now note that w^* matches all the constraints of (2.15). Indeed $w_{\epsilon_m}^1$ lies in the range of $\widehat{C}$ which is a closed space, also $v_{\epsilon_m} - \widehat{v} = w_{\epsilon_m}^0 + w_{\epsilon_m}^1 \underset{m \to \infty}{\longrightarrow} w^*$. It is clear that $v_{\epsilon_m} - \widehat{v}$ matches the equality and inequality constraints of (2.15) for all m and hence passing to the limit we have that w^* satisfies the equalities and inequalities.

From the uniqueness of the minimizer of (2.15) we have that w^* is equal to v^*. In particular this means that v^* is the unique cluster point of the original sequence

$(w^1_{\epsilon_m})_{m\in\mathbb{N}}$. Since the original sequence was arbitrarily chosen, we conclude that $\lim_{\epsilon\to 0} v_\epsilon = v^* + \widehat{v}$. □

Remark 2.4.3. Notice that the proof remains true if we take general convex inequalities. We simply need the constrained sets to be closed and convex; however we have restricted to linear equality and inequality constraints for simplicity and because these arise most often in practice.

2.5 Blood glucose forecasting experiments

Here we present an application of the constrained EnKF to the tracking and forecasting of human blood glucose levels. In Section 2.5, we use self-monitoring data collected by an individual with Type 2 Diabetes; in Section 2.5.2, we use electronic health record data from the Columbia University Irving Medical Center critical care unit.

We model the glucose-insulin system with the ultradian model proposed by Sturis et al. [398]. The primary state variables are the glucose concentration, G, the plasma insulin concentration, I_p, and the interstitial insulin concentration, I_i; these three state variables are augmented with a three stage delay (h_1, h_2, h_3) which encodes a non-linear delayed hepatic glucose response to plasma insulin levels. The resulting ordinary differential equations have the form:

$$\frac{dI_p}{dt} = f_1(G) - E(\frac{I_p}{V_p} - \frac{I_i}{V_i}) - \frac{I_p}{t_p} \tag{2.17a}$$

$$\frac{dI_i}{dt} = E(\frac{I_p}{V_p} - \frac{I_i}{V_i}) - \frac{I_i}{t_i} \tag{2.17b}$$

$$\frac{dG}{dt} = f_4(h_3) + m_G(t) - f_2(G) - f_3(I_i)G \tag{2.17c}$$

$$\frac{dh_1}{dt} = \frac{1}{t_d}(I_p - h_1) \tag{2.17d}$$

$$\frac{dh_2}{dt} = \frac{1}{t_d}(h_1 - h_2) \tag{2.17e}$$

$$\frac{dh_3}{dt} = \frac{1}{t_d}(h_2 - h_3) \tag{2.17f}$$

Here $m_G(t)$ represents a known rate of ingested carbohydrates, $f_1(G)$ represents the rate of glucose-dependent insulin production, $f_2(G)$ represents insulin-independent glucose utilization, $f_3(I_i)G$ represents insulin-dependent glucose utilization and $f_4(h_3)$ represents delayed insulin-dependent hepatic glucose production; the functional forms of these parameterized processes can be found in Section 2.5.3, along with a description of model parameters.

In the EnKF setting, we write $u = [I_p, I_i, G, h_1, h_2, h_3]$, and use (6.25) to define F such that

$$\frac{du}{dt} = F(u, t, \theta),$$

where θ contains model parameters. We then extend the state vector in order to perform joint parameter estimation: $v = [u, \theta]^T$.

For the purposes of this paper, the function $m_G(t)$ may be viewed as known; it is determined from data describing meals consumed by the patient. Since insulin (I_p and I_i) and delay variables (h_1, h_2, and h_3) are not measured, whilst glucose is measured, we define the measurement operator to be $H = [0, 0, 1, 0, 0, 0, 0]$. The discrete time forward model is obtained by integrating the deterministic model in (6.25) between consecutive measurement time-points and applying an identity map to θ. Because these time-points may not be equally spaced, and because the time-dependent forcings (meals) will differ in different time-intervals, this leads to a map of the form

$$v_{j+1} = \Psi_j(v_j).$$

This is a slight departure from the methodology outlined in Section 2.2, where Ψ does not depend on j (autonomous dynamics) but is a straightforward extension which the reader can easily provide.

We present EnKF results from a single patient's data when run with and without constraints (Algorithms 1 and 3 respectively). We performed joint state-parameter estimation, augmenting the state with parameter θ (see Section 2.5.3 for details of where specific parameters appear) and adding identity-map dynamics for parameter θ.

2.5.1 Type 2 diabetes self-monitoring data

We use the "P1" data set described by Albers et al. [10]; this dataset includes measurements of blood glucose and consumed nutrition, and is publicly available on `physionet.org`. For more information on the data, and on an unconstrained data assimilation approach using the unscented Kalman filter, see [10].

We choose to only estimate the parameter R_g (along with the physiologic state u), and impose the following constraints:

$$\begin{bmatrix} 0.01 \\ 0.01 \\ 2000 \\ 0.01 \\ 0.01 \\ 0.01 \\ 0 \end{bmatrix} \preceq v \preceq \begin{bmatrix} 10000 \\ 10000 \\ 40000 \\ 10000 \\ 10000 \\ 10000 \\ 1000000 \end{bmatrix} \tag{2.18}$$

Figure 2.1 compares the overall distribution of updated state means over time when running EnKF with and without these state constraints. While individual particles in this experiment often violated the constraints, the overall updated means did not. Nevertheless, enforcement of lower-bound constraints shifts up the state distribution slightly. Note that upper bound constraints were never violated in this experiment.

Figure 2.2 shows a two-dimensional state projection of updated particles at a given time step before and after applying the constrained optimization. Note that particles may additionally violate constraints in unplotted dimensions—this explains why one particle whose unconstrained update appears to live within the constraints is in fact differently updated under the constrained optimization. Time step 126 was selected for illustrative purposes, and was the measurement event in which particles most often violated the constraints.

Figure 2.3 depicts the overall frequency of constraint violations. We observe that the the measured state (blood glucose) never violated a constraint, nor did the inferred parameter R_g. However, other model states did often violate constraints, and up to 30% (4/13) of particles simultaneously violated the constraints at a single time-step.

By adding constraints, we ensure that all the simulations which constitute the ensemble method are biologically plausible.

2.5.2 Intensive care unit patient dataset

Here, we report results from [11], which applies the previously discussed modeling and filtering methodology to a dataset of blood glucose measurements and meal events collected in the neurological intensive care unit at the Columbia University Irving Medical Center. In this work, the unconstrained and constrained EnKF were compared across individual patient timeseries, and various physiologically motiviated constraint sets were further studied. They found the insulin state most necessary to

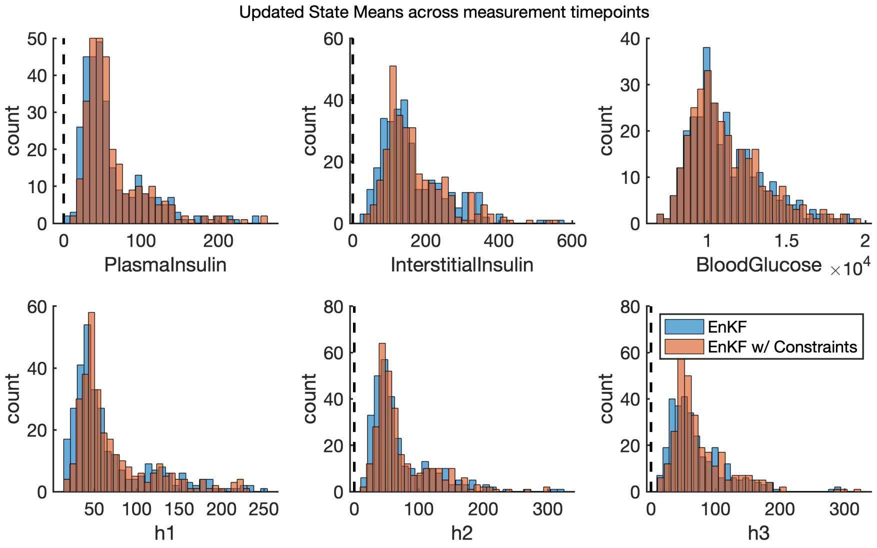


Figure 2.1: The distribution of mean state updates when running EnKF with and without inequality constraints. Black vertical lines denote lower bound state constraints.

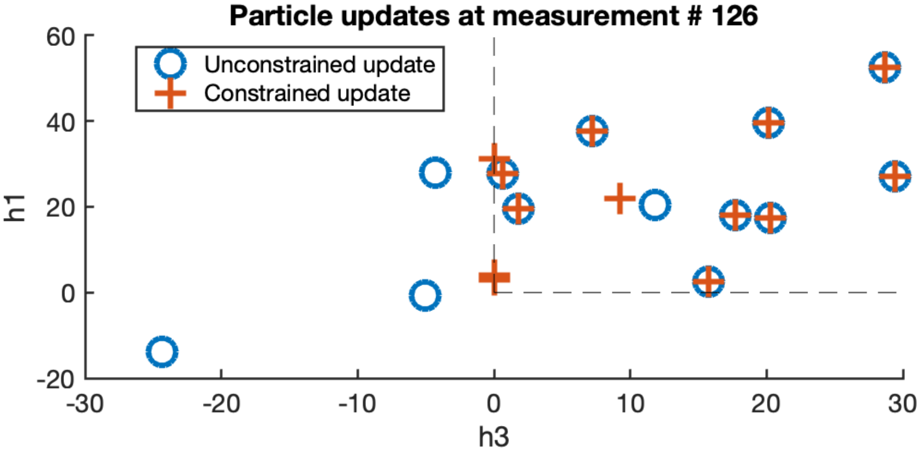


Figure 2.2: Particle updates at a given time-step (here, measurement 126) are shown using a traditional Kalman gain versus using the constrained optimization. The black lines denote lower bound constraints on the states h_1 and h_3.

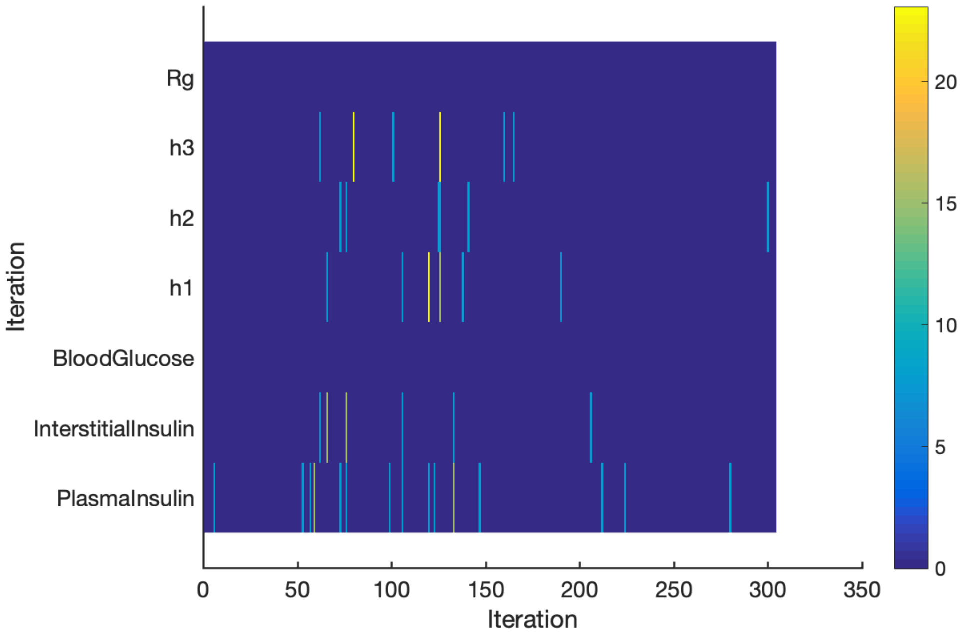


Figure 2.3: Percentage map of the constraint violations, where each lower-bound constraint is represented by a row. At each iteration, the percentage of particles that violated a constraint is color-coded, with yellow representing the largest proportion of constraint violations.

constrain, and found substantial success by imposing severe constraints on insulin levels ($I_i, I_p \in [75, 275]$) and not constraining any other state or parameter. The constrained EnKF (with the severe insulin constraints) enabled accurate forecasts after $1 - 1.5$ days of data instead of $4 - 4.5$ and decreased mean-squared-error by a factor of 5 when compared to the unconstrained EnKF approach.

To illustrate this improvement, we provide an example of glucose forecasting for patient 593 performed with and without insulin constraints. Figure 2.4 shows predictions and inferences without constraints—note that the insulin states eventually converge to the aforementioned constraint set, but spend significant time exploring physiologically unlikely values during the first 4 days of data collection. Figure 2.5 shows the same experiment performed with insulin constraints—we observe much faster convergence of the insulin states and faster synchronization with the observed glucose measurements, enabling predictions with significantly lower error to be available much sooner than in Figure 2.4.

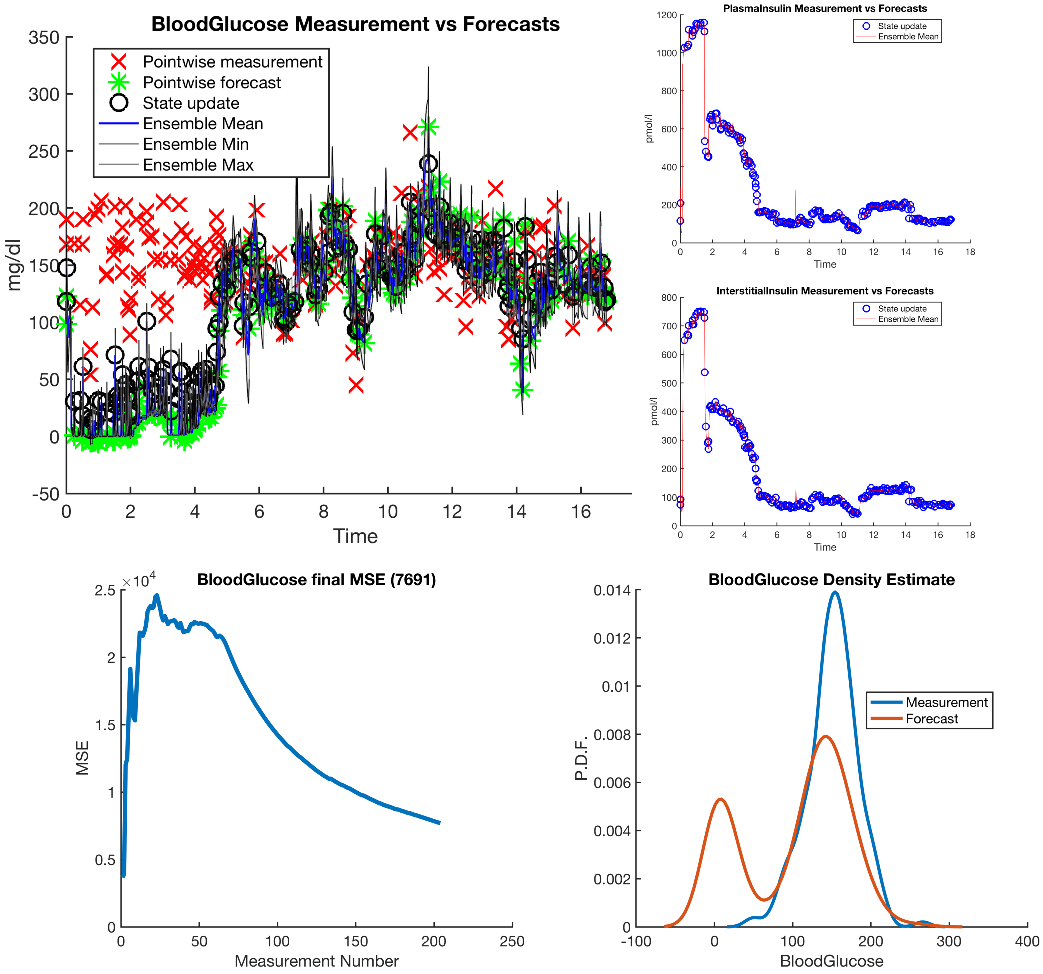


Figure 2.4: Oh caption, my caption.

2.5.3 Ultradian model of glucose-insulin dynamics

We give the details of the ultradian model of glucose-insulin dynamics used as the forward model in Section 2.5. An example of the induced dynamics is given in Figure 6.13.

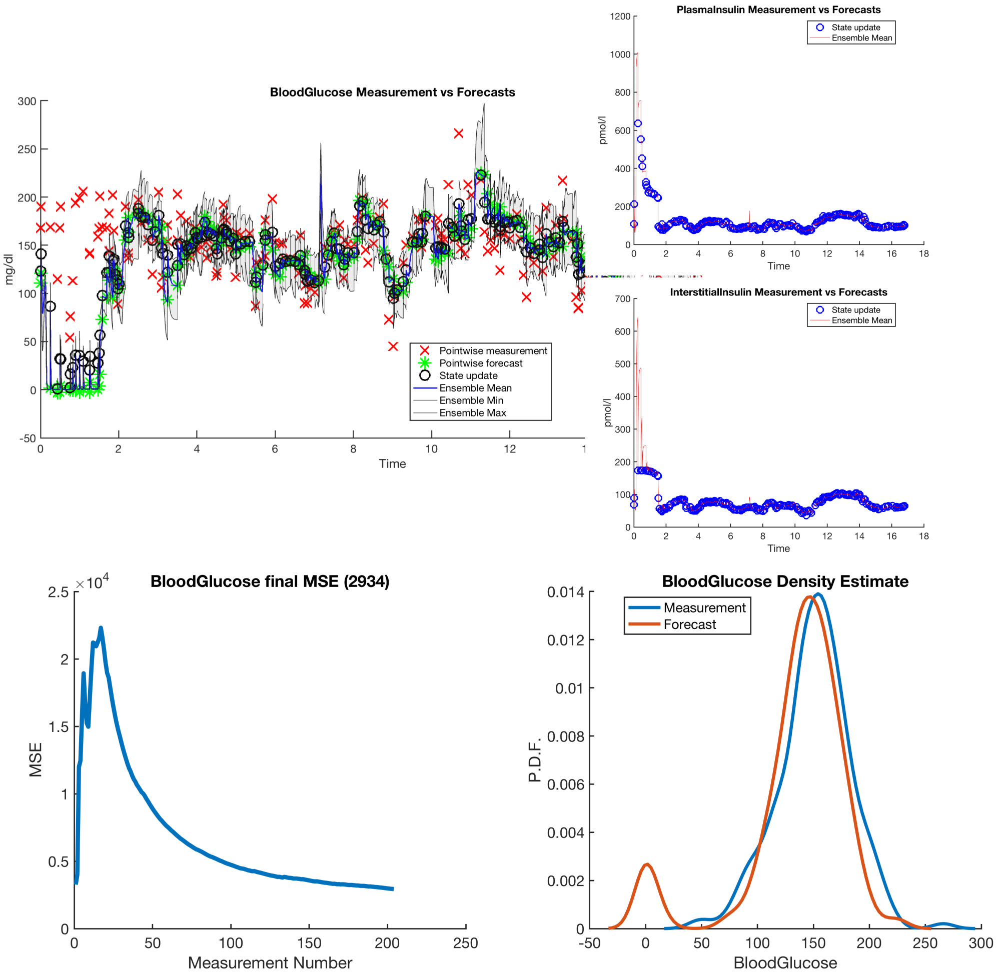


Figure 2.5: This figure demonstrates how constraints on the insulin states improve forecast accuracy and robustness, compared with the unconstrained EnKF forecast results shown in Figure 2.4. The right panels show insulin dynamics are quickly constrained to lie within the realistic constraint boundaries, resulting in the model entraining to the patient within 1 – 1.5 days instead of 4 – 4.5 days. The upper left panel shows the forecast ensemble converging and estimating the patient's mean BG; the variance in BG is under-estimated.

$$\frac{dI_p}{dt} = f_1(G) - E(\frac{I_p}{V_p} - \frac{I_i}{V_i}) - \frac{I_p}{t_p} \tag{2.19}$$

$$\frac{dI_i}{dt} = E(\frac{I_p}{V_p} - \frac{I_i}{V_i}) - \frac{I_i}{t_i} \tag{2.20}$$

$$\frac{dG}{dt} = f_4(h_3) + m_G(t) - f_2(G) - f_3(I_i)G \tag{2.21}$$

$$\frac{dh_1}{dt} = \frac{1}{t_d}(I_p - h_1) \tag{2.22}$$

$$\frac{dh_2}{dt} = \frac{1}{t_d}(h_1 - h_2) \tag{2.23}$$

$$\frac{dh_3}{dt} = \frac{1}{t_d}(h_2 - h_3) \tag{2.24}$$

where, for N meals at times $\{t_j\}_{j=1}^N$ with carbohydrate composition $\{m_j\}_{j=1}^N$

$$m_G(t) = \sum_{j=1}^{N} \frac{m_j k}{60} \exp(k(t_j - t)), \ \ N = \#\{t_j < t\} \tag{2.25}$$

and

$$f_1(G) = \frac{R_m}{1 + \exp(\frac{-G}{V_g c_1} + a_1)} \text{ : the rate of insulin production} \tag{2.26}$$

$$f_2(G) = U_b(1 - \exp(\frac{-G}{C_2 V_g})) \text{ : insulin-independent glucose utilization} \tag{2.27}$$

$$f_3(I_i) = \frac{1}{C_3 V_g}(U_0 + \frac{U_m - U_0}{1 + (\kappa I_i)^{-\beta}}), \ f_3(I_i)G \text{ : insulin-dependent glucose utilization} \tag{2.28}$$

$$f_4(h_3) = \frac{R_g}{1 + \exp(\alpha(\frac{h_3}{C_5 V_p} - 1))} \text{ : delayed insulin-dependent glucose utilization} \tag{2.29}$$

$$\kappa = \frac{1}{C_4}(\frac{1}{V_i} - \frac{1}{E t_i}) \tag{2.30}$$

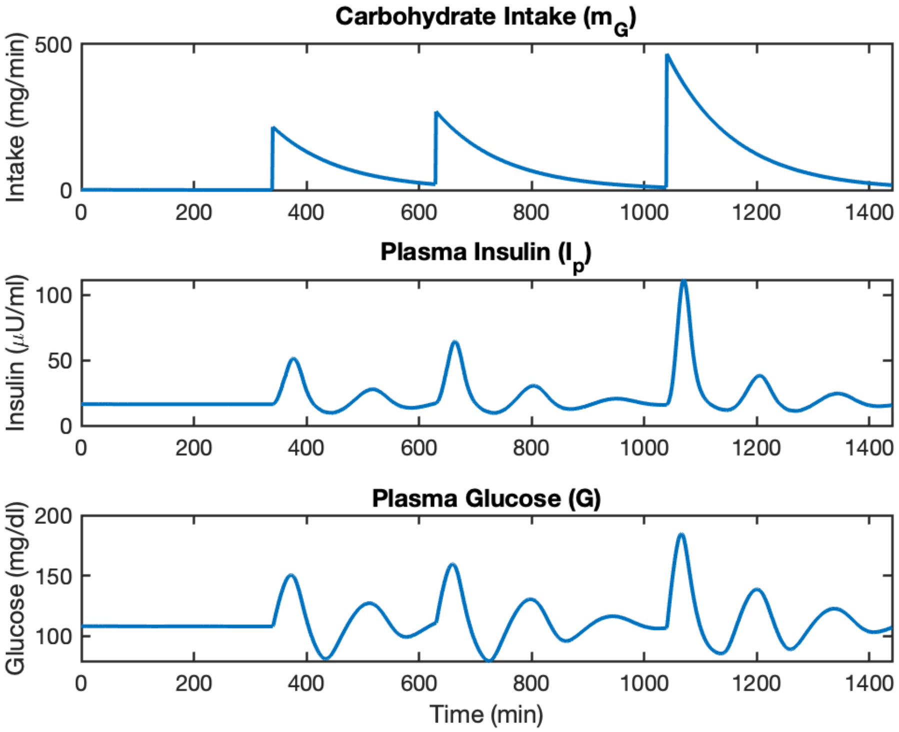


Figure 2.6: Here we show the oscillating dynamics of the glucose-insulin response in the ultradian model, driven by an exponentially decaying nutritional driver m_G.

Chapter 3

AN SDE-BASED REDUCED-ORDER MODEL OF GLUCOSE-INSULIN DYNAMICS

Remark 3.0.1. This chapter is derived from the manuscript by Sirlanci, Levine, Wang, Albers, and Stuart [389], which is under review at Chaos.

3.1 Introduction

Broadly speaking mathematical models of human physiology may serve one of two purposes: elucidation of the detailed mechanisms which comprise the complex systems underlying observed physiology; or prediction of outcomes from the complex system, for the purposes of medical intervention to ameliorate undesirable outcomes. In principle, these two objectives interact: a model which explains the detailed mechanisms, if physiologically accurate and compatible with observed data, will of course be good for prediction. However, human physiological data are often too sparse for use in resolving high-fidelity physiological details; moreover, this sparsity can induce severe model unidentifiability that impedes inference efficiency and results in suboptimal predictive performance. While the previous chapter focuses on methods for constraining inference, this chapter demonstrates how model reduction and stochastic closure techniques can be applied to physiologic models to make them more identifiable from available data. This, of course, comes with a cost of reduced fidelity; however, we find that this tradeoff often sides with model simplicity, especially when data are low-fidelity (i.e. sparse and noisy) and the underlying system is not fully understood (i.e. available ”high-fidelity" models have substantial inadequacies). The human glucose-insulin system provides an important example of this challenge because in many settings, insulin—a dominant state variable—is rarely measured.

The objective of the work presented here (and in [389]) is to distill existing mechanistic models of the human endocrine system into an interpretable model of human glucose dynamics that is identifiable from real-world clinical data. We do this by approximating the insulin's glycemic regulation as an Ornstein-Uhlenbeck process (a linear stochastic differential equation with exponential mean-reversion), then further introducing forcing terms that parameterize exogenous effects of nutrition and medication. We then evaluate the predictive performance of this simple model on clinical data sets in an outpatient type 2 diabetes setting and an inpatient intensive care unit setting. We compare its predictive performance with state-of-the-art predictions given by the constrained EnKF (Section 2.4) paired with popular mechanistic models of the glucose-insulin system (from which our reduced model was inspired).

The key finding from this work is *non-inferior* predictive capacity of our simple linear stochastic model when compared to higher complexity non-linear models. This indicates that the severity of our clinical data constraints prevented us from extracting additional expressivity from the non-linear models beyond the simple dynamics encoded by a forced linear SDE. Alternatively, it may be that the additional expressivity of the non-linear models is not of the right type, and thus does not offer much additional predictive advantage (despite having clear mechanistic validity).

3.1.1 Clinical settings

Type 2 Diabetes Mellitus (T2DM)

Intensive Care Units (ICU)

3.1.2 Advantages of a linear SDE model

In accordance with our goal, which is to develop a highly simplified yet interpretable model, we work with a forced SDE of Ornstein-Uhlenbeck type to describe glucose evolution, together with an observation model of linear form, subject to additive Gaussian noise. The Gaussian structure allows for computational tractability in prediction since probability distributions on the glucose state are described by Gaussians and hence represented by simply a mean and variance. Note that the protocols for managing glucose depend on intervals; e.g., a goal may be to keep glucose between 80-150 mg/dl and interval deviation from this goal, e.g., 151-180 mg/dl, induce changes in the insulin dosage. This means that decisions are made based on boundaries of glycemic trajectories. Nevertheless, because glucose oscillates under continuous feeding, clinicians typically aim to ensure that the glycemic mean does not fall below 60 mg/dl or above 180 mg/dl for any length of time. The intervals are then a proxy for this balance of managing the mean and protecting against trajectories diverging too high or low at any time, including between observations. Hence accurately resolving mean and standard deviation in BG levels is extremely important.

Furthermore the Kalman filter may be used to incorporate the data, and also works entirely within the Gaussian framework; and finally parameter learning, although non-Gaussian, is well-developed in the Kalman filter setting, both from the fully

Bayesian and optimization (empirical Bayes) perspectives. The Ornstein-Uhlenbeck process has three contributions: a damping term which drives the BG level towards its base value at a rate which is possibly insulin dependent; a forcing representing nutritional intake and a white noise contribution, which is used to encapsulate the high-frequency dynamics as these dynamics are difficult to be resolved with sparse measurements. The presence of noise in the glucose evolution model, as well as in the data acquisition process, allows for model error which is natural in view of the the rather simple modeling framework.

Our Contribution

- We describe a simple, interpretable, modeling framework limited to states and parameters that are directly observable or inferable from data for prediction within the human glucose-insulin system, based on a continuous time linear, Gaussian, stochastic differential equation (SDE) for glucose dynamics, in which the effect of insulin appears parametrically.
- We completely describe and detail the inference machinery necessary—in a data assimilation and inverse problems framework—to estimate a stochastic differential equation model of glucose dynamics with real-world data.
- The framework is sufficiently general to be usable within the ICU, T2DM and potentially T1DM settings.
- We demonstrate, in a train-test set-up, that the models are able to fit individual patients with reasonable accuracy; both ICU and T2DM data are used. The test framework we use is a predictive one laying the foundations for future control methodologies.
- Comparison of the data fitting for T2DM and ICU patients reveals interesting structural differences in their glucose regulation.
- We make a comparison of the predictive power of our stochastic modeling framework with that of more sophisticated models developed for both T2DM and the ICU, demonstrating that the simple stochastic approach is generally more accurate in both settings.

Outline

An outline of the chapter is as follows. In Section 3.3, we introduce the general continuous-time mathematical model that describes the human glucose-insulin system. Then, in Section 3.3.2, we introduce the specific versions of this model relevant in T2DM and ICU settings. The two model classes all derive from a single general model, and differ according to how nutrition uptake and glucose removal are represented. In Section 3.4, we construct the framework for stating the parameter estimation problem and its solution. In Section 3.5, we describe the datasets, the experiments we design for parameter estimation and forecasting, and the methods we use for parameter estimation and forecasting for the T2DM and ICU settings. Section 3.6 presents the numerical results on parameter estimation and forecasting along with some uncertainty quantification (UQ) results separately for T2DM and ICU settings. Finally, in Section 3.7, we make some concluding remarks and discuss future directions that we intend to pursue.

3.2 Literature Review

Researchers have developed various mathematical models ranging from extremely simple to highly complex, using ordinary differential equations (ODEs) and machine learning (ML) to predict and describe human glucose metabolism. We discuss these efforts organized according to model usage.

Models Developed to Understand Physiology, Pathophysiology and Disease Pathogenesis: Some mechanistic models are developed to investigate a specific phenomenon of the glucose-insulin system such as to understand the different phases of insulin secretion with respect to different glucose stimulation patterns, to estimate insulin sensitivity in the intravenous glucose tolerance test (IVGTT) setting, and to elucidate the cause of the ultradian (long-period) oscillations of insulin and glucose, [156, 35, 240, 384, 399, 239, 259]. Others have developed models by clinically minded motivations to describe β-cell mass, glucose, and insulin dynamics and to investigate T2DM pathophysiology, [407, 36, 158, 146]. Some researchers developed models to describe the underlying system in more detailed way such as the events that occur during oral glucose ingestion [98, 231], or relevant organ systems, [121]. A nice review of the models developed for clinical and physiological investigation BG homeostasis and T2DM can be found in [280].

There are also machine learning models developed to understand model phenotypic and health care process differences and to predict T2DM development, [3, 12, 7, 6, 183, 186, 10, 1, 197].

Models Developed for Prediction and Control: Researchers have developed mechanistic models to address challenges including fast evolution of the underlying system (parameter variation in time), wide variation in clinical response within and between patients, sparse measurements, and concerns about safety issues with the goals of prediction and control of BG levels, [379, 414, 248, 251, 328, 211, 357, 212, 419, 169, 314]. Others developed stochastic (mechanistic) models with the same purpose, [457, 102, 250, 249, 228, 103, 119].

Glucose control based on mechanistic modeling is the focus of the artificial pancreas project in the type 1 diabetes mellitus (T1DM) setting and many models are developed for this purpose, [58, 312, 92, 130, 129, 216, 313]. A comprehensive range of BG control algorithms can be found in [76]. Finally, other researchers conducted clinical trials to compare the efficacy between different closed-loop artificial pancreas systems and sensor-assisted pump therapy for T1DM patients, [31, 57, 62, 404, 406].

ML approaches have been proposed in pure prediction tasks such as predicting next glucose values or hypoglycemia. For these purposes, some researchers used classification methods and neural network models, [401, 293, 144, 450, 39, 356, 464], while others used ARIMA (auto-regressive integrated moving average) and linear regression models, [281, 443, 292, 347, 52, 141, 393, 458].

Finally, in [286], the authors developed a hybrid model balancing a physiological and statistical model of glucose-insulin dynamics to forecast long-term BG levels of T1DM patients based on real-world data, showing the possibility of outperforming the forecasting of BG levels obtained by either pure physiological or pure statistical models alone.

Models Used for Patient-Centered Disease Self-Management: Patient-centered disease self-management is a crucial tool to improve health condition of patients focusing on their needs, life style, and preferences. Some researchers developed decision support tools for T2DM patients based on mechanistic or machine learning models, [276, 277, 106, 105, 8, 290].

Mathematical Techniques Used for Estimation: In all of the models discussed above, parameter estimation plays a vital role in the accuracy of predictions. Parameters are rarely directly measurable, and their values will vary from one patient to another. There are two overarching approaches to estimating parameters, optimization where a model-data mismatch is minimized to determine parameters [124], and the Bayesian approach [199] where the distribution of the parameters, given the data and given the assumed (noisy) model-data framework, is computed. Researchers used various approaches for parameter estimation. The most common approaches are the standard least squares optimization, [414, 441], nonlinear least squares optimization, [173], and Bayesian approach to estimate both time-invariant and time-varying model parameters, [181].

3.3 Modeling framework

We begin by providing a constructive explanation of the continuos-time model including the model equations, the description of unknown model parameters and the precise role of each component of the model. Then, we also provide a detailed conceptual explanation of how a stochastic modeling approach could be used to represent blood glucose dynamics.

3.3.1 Model construction

To begin construction of a simple, one-state model for glucose dynamics, we first consider the classical two-state Bergman [35] equations:

$$\dot{G} = m_{\text{external}}(t) + f_{\text{HGP}}(G) - (c + s_I I)G \tag{3.1a}$$

$$\dot{I} = I_{\text{external}}(t) + \beta f_{\text{ISR}}(G) - kI. \tag{3.1b}$$

Here, G denotes plasma glucose concentration and I denotes plasma insulin concentration. External inputs of nutrition and inuslin are given by $m_{\text{external}}(t)$, $I_{\text{external}}(t)$, respectively. The insulin dynamics, beyond external forcing, are primarily governed by a glucose-dependent secretion rate $f_{\text{ISR}}(G)$, insulin-producing beta-cell mass β, and linear degradation rate k. The glucose dynamics, aside from external forcing (i.e. meals), are driven by a glucose-dependent (insulin independent) hepatic glucose production f_{HGP}, an insulin-dependent glucose removal rate IG (with insulin sensitivity factor s_I), and a linear degradation rate c.

In this work, we hypothesize that the pancreatic and hepatic regulation of glucose can instead be approximated by a simple function of glucose $f_{\text{internal}}(G)$ and a closure term ν. This results in a new single-state equation

$$\dot{G} = m_{\text{external}}(t) + f_{\text{internal}}(G) + f_{\text{external}}(I) + \nu(t), \tag{3.2}$$

where the closure term $\nu(t)$ accounts for additional glycemic dynamics not captured by the first three terms. To begin evaluating the utility of this perspective, we choose simple forms for these unknown functions.

Specifically, we assume that glucose regulation can be roughly approximated by an exponential decay to a fixed point G_b at rate γ such that $f_{\text{internal}}(G) := \gamma\big(G_b - G(t)\big)$.

We also assume that the effect of external insulin delivery has a simple relationship $f_{\text{external}}(I) := \beta I_{\text{external}}$ with proportionality constant β. Finally, we assume that the possibly large residual errors induced by these simplifiying assumptions are given by a Brownian Motion $W(t)$ with variance proportional to γ (with proportionality constant σ^2); i.e $v(t) := \sqrt{2\gamma\sigma^2}\dot{W}(t)$.

These choices yield the following Ornstein-Uhlenbeck model for evolution of blood glucose $G(t)$:

$$\dot{G}(t) = -\gamma(G(t) - G_b) + m(t) - \beta I(t) + \sqrt{2\gamma\sigma^2}\dot{W}(t). \tag{3.3}$$

There are four basic parameters for the model in eq. (3.3). G_b (mg/dl) represents the basal glucose (i.e. the mean of the unforced process), γ (1/min) is the decay rate for the exponential mean reversion, β (mg/(dl*U)) is a proportionality constant for the linear effect of IV insulin-based glucose removal, and σ governs the variance of the oscillations described by $W(t)$.

We use simple models for the meal function $m(t)$ and the insulin delivery function $I(t)$ (defined in section 3.3.2) that enable explicit solution of the continuous time model between events. We define *events* as times at which the meal or insulin delivery functions change discontinuously, or points at which BG is measured.

The simple linear Gaussian structure of Ornstein-Uhlenbeck models, along with appropriately simple forcing terms $m(t), I(t)$ (defined in section 3.3.2), allow for tractable solutions to eq. (3.3). Specifically, integration of the system leads to a solution $G(t)$ that is normally distributed with analytically calculable means and variances.

3.3.2 Event-Time Model

For computational purposes, and because data are typically available at discrete times, we develop a discrete-time version of the model (3.3). We first present it in generality, then develop it specifically for outpatient Type 2 Diabetes (T2DM) glucose modeling (see Section 3.3.3) and for inpatient intensive care unit (ICU) glucose modeling (see Section 3.3.4). Note that ICU and T2DM settings are also the focus for our data-driven studies.

The time discretization is defined completely by a dataset in the following sense. Let $\{t_k^{(m)}\}_{k=1}^{K_m}$ denote the times of relevant nutrition events, let $\{t_k^{(i)}\}_{k=1}^{K_i}$ denote the times of relevant insulin delivery events, and let $\{t_k^{(o)}\}_{k=1}^{K_o}$ denote the times of glucose measurements. We call the re-ordered union of these sets,

$$\{t_k\}_{k=0}^{N} := \{t_k^{(m)}\}_{k=1}^{K_m} \cup \{t_k^{(i)}\}_{k=1}^{K_i} \cup \{t_k^{(o)}\}_{k=1}^{K_o}$$

as *event times*.

We can obtain the following *event-time model* by integrating (3.3) over the event-time intervals, $[t_k, t_{k+1})$ for $k = 0, 1, ..., N-1$, via use of Itô formula: [1]

$$G(t_{k+1}) = G_b + e^{-\gamma h_k}(G(t_k) - G_b) + \int_{t_k}^{t_{k+1}} e^{-\gamma(t_{k+1}-s)} m(s)ds - \int_{t_k}^{t_{k+1}} e^{-\gamma(t_{k+1}-s)} I(s)ds + \sigma\sqrt{1 - e^{-2\gamma h_k}}\xi_k, \tag{3.4}$$

where $h_k := t_{k+1} - t_k$ and $\xi_k \sim N(0,1)$ independent random variables. We exhibit specific versions of this general event-time model for T2DM and ICU settings in more detail in the following sections.

3.3.3 T2DM

Glucose dynamics in type 2 diabetes settings are driven by a combination of diet, activity, medication, and internal physiology. Here, we specifically focus on modeling the effect of carbohydrate intake on glycemic levels of people with T2DM. Because our T2DM self-monitoring datasets are collected from patients who do not take insulin, we have $I(t) \equiv 0$ for this setting, and will thus ignore the exogenous insulin term in the T2DM event-time model. The meal function, $m(t)$, on the other hand, is essential for capturing the uptake of glucose into the bloodstream from consumed carbohydrates. Here, we define $m(t)$ as the difference of two exponential functions (this choice was shown to be effective in the T2DM case by Albers et al. [10]):

$$m(t) = \sum_{k=1}^{K_m} \frac{G_k}{c_k}\left(e^{-a(t-t_k^{(m)})} - e^{-b(t-t_k^{(m)})}\right)\mathbb{1}_{[t_k^{(m)},\infty)}(t) \tag{3.5}$$

where $t_k^{(m))}$ is the time of the k^{th} meal, G_k is the total amount of glucose (mg) in the k^{th} meal, and c_k (/dl) is a normalizing constant so that $\int_{t_k^{(m)}}^{\infty}(e^{-a(t-t_k^{(m)})} - e^{-b(t-t_k^{(m)})})dt = 1$. Therefore, the model in (3.3) becomes

$$\dot{G}(t) = -\gamma(G(t) - G_b) + \sum_{k=1}^{K_m} \frac{G_k}{c_k}\left(e^{-a(t-t_k^{(m)})} - e^{-b(t-t_k^{(m)})}\right)\mathbb{1}_{[t_k^{(m)},\infty)}(t) + \sqrt{2\gamma\sigma^2}\dot{W}(t), \tag{3.6}$$

[1] equivalent to using integrating factors in this case

in the T2DM setting. In this model, the first term represents body's own effect to remove insulin from the bloodstream, the second term represents the effect of nutrition on the rate of change of BG, and the last term models the residual infidelities of our model as a Brownian Motion. Integrating over $[t_0, t]$, we can write the analytic solution of this equation as

$$\begin{aligned} G(t) &= G_b + e^{-\gamma(t-t_0)}(G(t_0) - G_b) \\ &\quad + \sum_{k=1}^{K_m} \frac{G_k}{c_k}\left(\frac{e^{-a(t-t_k^{(m)})} - e^{-\gamma(t-t_k^{(m)})}}{\gamma - a} - \frac{e^{-b(t-t_k^{(m)})} - e^{-\gamma(t-t_k^{(m)})}}{\gamma - b}\right)\mathbb{1}_{[t_k^{(m)},\infty)}(t) \\ &\quad + \int_{t_0}^{t} e^{-\gamma(t-s)}\sqrt{2\gamma\sigma^2}dW(s). \end{aligned} \tag{3.7}$$

Note that, in practice, we need to evaluate BG level at specific time points and hence need the discrete-time model implied by the continuous time representation in (3.7). Now, by integrating (3.6) over $[t_k, t_{k+1})$ and denoting $g_k := G(t_k)$, we obtain

$$g_{k+1} = G_b + e^{-\gamma h_k}(g_k - G_b) + m_k + \sigma\sqrt{1 - e^{-2\gamma h_k}}\xi_k, \tag{3.8}$$

as a special case of (3.4). Also, for any fixed t_k, find the meal times $t_j^{(m)}$ such that $t_j^{(m)} \leq t_k$ and denote the index set of these meal times by $\mathcal{I}_k$. Then m_k in (3.8) becomes

$$m_k = \sum_{j\in\mathcal{I}_k} \frac{G_j}{c_j}\left(\frac{e^{-a(t_{k+1}-t_j^{(m)})} - e^{-\gamma h_k}e^{-a(t_k-t_j^{(m)})}}{\gamma - a} - \frac{e^{-b(t_{k+1}-t_j^{(m)})} - e^{-\gamma h_k}e^{-b(t_k-t_j^{(m)})}}{\gamma - b}\right). \tag{3.9}$$

Hence, note that in this case, we have five model parameters to be estimated: $G_b, \gamma, \sigma, a, b$. Recall that in this setting, G_b represents the basal glucose value that BG level stays around starting some time after nutrition intake until the next nutrition intake. γ represents the decay rate of BG level to G_b after the nutrition intake, and σ represents the amplitude of the BG level oscillations. The parameters a and b entering the meal function implicitly control the time needed for the glucose nutrition rate to reach its peak value, and the time needed for this rate to return back to the vicinity of 0. Because of these simple physiological meanings, the parameters entering the event-time model are important not only for accurately capturing, and predicting, glucose dynamics based on data, but also contain implicit information about the health condition of the patient. For example, the basal glucose value is measured during some tests to check if an individual is healthy, pre-diabetic, or diabetic.

3.3.4 ICU

In the ICU setting, glucose dynamics are given by a combination of changing patient physiologic state, nutrition (delivered intravenously and enterally through a feeding tube that runs to the gut), and insulin delivery. In our ICU dataset, 8-10% of the ICU patients are diabetic and only 5% of those are T1DM patients. However, more than 90% of ICU patients require glycemic management and 10-20% of them experience a hypoglycemic event over the course of management. Consequently, regardless of being diabetic or non-diabetic, they are typically given IV insulin to control BG levels. Thus, we must model both $m(t)$ and $I(t)$ in this setting.

We choose to model these external forcings as piecewise constants functions; this choice correspondes to clinical practice, in which constant infusions are peridiocally adjusted, and also allows for simple calculations. Here, we define the nutritional forcing function as:

$$m(t) = \sum_{k=1}^{K_m} d_k \mathbb{1}_{[t_k^{(m)}, t_{k+1}^{(m)})}(t), \tag{3.10}$$

where $t_k^{(m)}$ is the time at which a clinician changes the nutrition delivery rate, d_k is the nutrition rate over the time interval $[t_k^{(m)}, t_{k+1}^{(m)})$; these features are both directly available in our clinical dataset.

Similarly, we define the external insulin delivery rate as:

$$I(t) = \sum_{k=1}^{K_i} i_k \mathbb{1}_{[t_k^{(i)}, t_{k+1}^{(i)})}(t), \tag{3.11}$$

where i_k is the rate of insulin over the time interval $[t_k^{(i)}, t_{k+1}^{(i)})$, again obtained directly from the data set.

Therefore, substituting (3.10) and (3.11) into the general equation (3.3), the ICU version of our model becomes

$$\dot{G}(t) = -\gamma(G(t) - G_b) + \sum_{k=1}^{K_m} d_k \mathbb{1}_{[t_k^{(m)}, t_{k+1}^{(m)})}(t) - \beta \sum_{k=1}^{K_i} i_k \mathbb{1}_{[t_k^{(i)}, t_{k+1}^{(i)})}(t) + \sqrt{2\gamma\sigma^2}\dot{W}(t). \tag{3.12}$$

In this model, the first term models the glucose removal rate with body's own effort (γ), the second term shows the effect of nutrition $m(t)$ on the BG level, the third term, $\beta I(t)$, models the external insulin effect, and the last term models infidelities as Brownian Motion.

We integrate (3.12) to get the analytical solution for any $t \geq t_0$ as follows

$$G(t) = G_b + e^{-\gamma(t-t_0)}(G(t_0) - G_b) + \sum_{k=1}^{K_m} d_k \int_{t_0}^{t} e^{-\gamma(t-s)} \mathbb{1}_{[t_k^{(m)}, t_{k+1}^{(m)})}(s) ds$$
$$- \beta \sum_{k=1}^{K_i} i_k \int_{t_0}^{t} e^{-\gamma(t-s)} \mathbb{1}_{[t_k^{(i)}, t_{k+1}^{(i)})}(s) ds + \sqrt{2\gamma\sigma^2} \int_{t_0}^{t} e^{-\gamma(t-s)} dW(s). \tag{3.13}$$

As in the previous section, we can also integrate (3.12) over $[t_k, t_{k+1})$ to obtain solutions at event-times

$$g_{k+1} = G_b + e^{-\gamma h_k}(g_k - G_b) + \frac{1}{\gamma}(1 - e^{-\gamma h_k})d_k - \beta \frac{1}{\gamma}(1 - e^{-\gamma h_k})i_k + \sigma\sqrt{1 - e^{-2\gamma h_k}}\xi_k \tag{3.14}$$

as another special case of (3.4). Here, we have four model parameters to estimate: $G_b, \gamma, \sigma, \beta$. Remember once again, G_b is the basal glucose value and γ is the decay rate of the BG level to its basal value, and σ is a measure for the magnitude of the BG oscillations. Finally, β is another proportionality constant, which is used to scale the effect of IV insulin on the BG rate change appropriately. These four parameters represent physiologically meaningful quantities that could properly resolve the mean and variance of the BG level.

3.4 Parameter Estimation

Our goal in this section is to formulate the parameter estimation problem. In Section 3.4.1, we construct an overarching Bayesian framework for our parameter estimation problems. We then describe two solution approaches for this problem: an optimization based approach which identifies the most likely solution, given our model and data assumptions; and MCMC, which samples the distribution on parameters, given data, under the same model and data assumptions. These two solution approaches are detailed in Sections 3.4.2 and 3.4.3, respectively.

As shown in detail before, our model takes slightly different forms in the T2DM and ICU settings. In the former the model parameters to be estimated are $G_b, \gamma, \sigma, a, b$ whereas in the latter the unknown parameters are $G_b, \gamma, \sigma, \beta$. However, we adopt a single approach to parameter estimation. To describe this approach we let the vector, θ represent the unknown model parameters to be determined from the data, noting that this is a different set of parameters in each case. Many problems in biomedicine, and the problems we study here in particular, have both noisy models and noisy data, leading to a relationship between parameter θ and data y of the form

$$y = \mathcal{G}(\theta, \zeta) \tag{3.15}$$

where unknown ζ is a realization of a mean zero random variable, but its value is not known to us. The objective is to recover θ from y. We will show how our models of the glucose-insulin system lead to such a model.

3.4.1 Bayesian Formulation

The Bayesian approach to parameter estimation is desirable for two primary reasons: first it allows for seamless incorporation of imprecise prior information with uncertain mathematical model and noisy data, by adopting a formulation in which all variables have probabilities associated to them; secondly it allows for the quantification of uncertainty in the parameter estimation. Whilst extraction of information from the posterior probability distribution on parameters given data is challenging, stable and practical computational methodology based around the Bayesian formulation has emerged over the last few decades; see [396]. In this work, we will follow two approaches: (a) obtaining the *maximum a posteriori (MAP) estimator*, which leads to an optimization problem for the most likely parameter given the data, and (b) obtaining *samples* from the posterior distribution on parameter given data, using Markov Chain Monte Carlo (MCMC) techniques.

Now let us formulate the parameter estimation problem. Within the event-time framework, let $g = [g_k]_{k=0}^N$ be the vector of BG levels at event times $\{t_k\}_{k=0}^N$, and $y = [y_k]_{k=1}^{K_o}$ be the vector of measurements at the measurement times $\{t_k^{(o)}\}_{k=1}^{K_o} \subset \{t_k\}_{k=0}^N$. By using the event-time version, and defining $\{\xi_k\}_{k=0}^N$ to be independent and identically distributed standard normal random variables, we see that given the parameters θ, g has multivariate normal distribution, i.e., $\mathbb{P}(g|\theta) = N(m(\theta), C(\theta))$. Equivalently,

$$g = m(\theta) + \sqrt{C(\theta)}\xi, \quad \xi \sim N(0, I). \tag{3.16}$$

Let L be a $K_o \times (N+1)$ matrix that maps $\{g_k\}_{k=0}^N$ to $\{y_k\}_{k=1}^{K_o}$. That is, if a measurement $i \in 1, ..., K_o$ is taken at the event time t_j, $j \in 0, 1, ..., N$, then the i^{th} row of L has all 0's except the $(j+1)^{st}$ element, which is 1. Adding a measurement noise, we state the observation equation as follows:

$$y = Lg + \sqrt{\Gamma(\theta)}\eta, \quad \eta \sim N(0, I), \tag{3.17}$$

where $\Gamma(\theta)$ is a diagonal matrix representing the measurement noise. Thus, we obtain the likelihood of the data, given the glucose time-series and the parameters, namely

$$\mathbb{P}(y|g,\theta) = N(Lg, \Gamma(\theta)).$$

However, when performing parameter estimation, we are not interested in the glucose time-series itself, but only in the parameters. Thus we directly find the likelihood of the data given the parameters (implicitly integrating out g) by combining (3.16) and (3.17) to obtain

$$y = Lm(\theta) + \sqrt{S(\theta)}\zeta, \quad \zeta \sim N(0, I), \tag{3.18}$$

where $S(\theta) = LC(\theta)L^T + \Gamma(\theta)$. Since ζ has multivariate normal distribution, using the properties of this distribution, we find that given the parameters, θ, y also has multivariate normal distribution with mean $Lm(\theta)$ and covariance matrix $S(\theta)$. This is the specific instance of equation (3.15) that arises for the models in this paper.

We have thus obtained $\mathbb{P}(y|\theta) = N(Lm(\theta), S(\theta))$, that is,

$$\mathbb{P}(y|\theta) = \frac{1}{\sqrt{(2\pi)^{K_m}\det(S(\theta))}} \exp\left(-\frac{1}{2}(y - Lm(\theta))^T S(\theta)^{-1/2}(y - Lm(\theta))\right); \tag{3.19}$$

this is the likelihood of the data, y, given the parameters, θ. Also, since we prefer to use $-\log(\mathbb{P}(y|\theta))$ rather than directly using $\mathbb{P}(y|\theta)$ for the sake of computation, we state it explicitly as follows:

$$-\log(\mathbb{P}(y|\theta)) = \frac{K_m}{2}\log(2\pi) + \frac{1}{2}\log(\det(S(\theta))) + \frac{1}{2}(y - Lm(\theta))^T S(\theta)^{-1}(y - Lm(\theta)). \tag{3.20}$$

Moreover, by using Bayes Theorem, we write

$$\mathbb{P}(\theta|y) = \frac{\mathbb{P}(y|\theta)\mathbb{P}(\theta)}{\mathbb{P}(y)} \propto \mathbb{P}(y|\theta)\mathbb{P}(\theta). \tag{3.21}$$

Note that the second statement of proportionality follows from the fact that the term, $\mathbb{P}(y)$, on the denominator is constant with respect to the parameters, θ, and plays the role of a normalizing constant.

From another point of view, considering (3.16) and (3.18), we see that given θ, (g, y) has multivariate normal distribution with mean and covariance matrix that could be computed from the above equations since, given θ, everything is explicitly known. Then, integrating g out, in other words, computing the marginal distribution we obtain the distribution of $y|\theta$, which corresponds to the one stated in (3.18).

Now, to define the prior distribution $\mathbb{P}(\theta)$ we assume that the unknown parameters are distributed uniformly across a bounded set Θ and define

$$\mathbb{P}(\theta) = \frac{1}{|\Theta|}\mathbb{1}_{\Theta}(\theta) = \begin{cases} \frac{1}{|\Theta|}, & \theta \in \Theta, \\ 0, & \theta \notin \Theta, \end{cases} \tag{3.22}$$

where $\mathbb{1}_{\Theta}(\cdot)$ is the characteristic function and $|\Theta|$ is the volume of the region defined by Θ. Thus, by substituting the likelihood, (3.19), and the prior distribution, (3.22), into (3.21), we formulate the posterior distribution as follows

$$\mathbb{P}(\theta|y) = \frac{1}{|\Theta|\sqrt{(2\pi)^{K_m}\det(S(\theta))}} \exp\left(-\frac{1}{2}(y - Lm(\theta))^T S(\theta)^{-1/2}(y - Lm(\theta))\right)\mathbb{1}_{\Theta}(\theta). \tag{3.23}$$

Now, we will show how we use this posterior distribution to state the parameter estimation problem.

3.4.2 Optimization

In this approach, the goal is to determine parameter values, θ, which maximize the posterior distribution, $\mathbb{P}(\theta|y)$ and is called to be the *MAP estimator*. Using the prior distribution as specified above, the parameter estimation problem becomes

$$\theta^* = \arg\max_{\theta} \mathbb{P}(\theta|y) = \arg\max_{\theta\in\Theta} \mathbb{P}(y|\theta) = \arg\min_{\theta\in\Theta} -\log(\mathbb{P}(\theta|y)). \tag{3.24}$$

Then, substituting (3.20) into (3.24), the problem will take the form

$$\theta^* = \arg\min_{\theta\in\Theta} ||S(\theta)^{-1/2}(y - Lm(\theta))||^2 + \log(\det(S(\theta))). \tag{3.25}$$

Hence, placing uniform prior distribution turns the problem of finding the MAP estimator into a constrained optimization problem. To solve this problem, we use built-in `MATLAB` functions, such as `fmincon` and `multistart`. `fmincon` is a gradient-based minimization algorithm for nonlinear functions. `multistart` starts the optimization procedure from the indicated number of starting points that are picked uniformly over the region defined by the constraints. It uses `fmincon` and other similar type of algorithms to perform each optimization process independently and provides the one that achieves the minimum function value among the result of all separate runs. With this approach, we have the opportunity to compare different optimization procedures that starts from different initial points. This provides some intuitive understanding of the solution surface and hence the estimated optimal parameters.

3.4.3 MCMC

Once an optimal point has been found, we may also employ the Laplace approximation [275, 305] to obtain a Gaussian approximation to the posterior distribution. The Laplace approximation is a reasonable approximation in many data rich scenarios in which all parameters are identifiable from the data, because of the Bernstein Von Mises Theorem [413], which asserts that the posterior distribution will then be approximately Gaussian, centered near the truth and with variance which shrinks to zero as more as more data is acquired. However data is not always abundant, and not all parameters are identifiable even if it is; in this setting sampling the posterior distribution is desirable. MCMC methods are a flexible set of techniques which may be used to sample from a target distribution, which is not necessarily analytically tractable, [258, 352]. For example, the distribution $\mathbb{P}(\theta|y)$ is the conditional distribution of the random model parameters, θ given the data, y. Even though we can explicitly formulate it as in equation (3.21), it is not always an easy task to extract useful quantities, such as posterior mean and variance, from that formula. In such cases, MCMC techniques are used to generate random samples from this target distribution and this random sample is used to obtain the desired information, which could be anything such as the mean, mode, covariance matrix, or higher moments of the parameters. Moreover, this technique is also very helpful to obtain UQ results for the estimated parameters.

In order to obtain more extensive knowledge than MAP estimator can provide about the posterior distribution of parameters given the data, $\theta|y$, we use MCMC methods as a natural choice to sample from that distribution. Among different possible algorithms (see [143]), we use the standard random walk Metropolis-Hastings algorithm. In order to make sure the resulting sample is indeed a good representer of the posterior distribution, we perform some diagnostics such as checking if chains for each parameter converged and if they are uncorrelated. Then, after removing the burn-in period, we compute the mean and the covariance matrix fro the remaining part of the sample. We use the mean as a point estimator for simulation and forecasting, and the covariance matrix provided valuable information to quantify uncertainty for the estimated parameters.

In practice, it can be hard to obtain efficient results with MCMC methods even when sampling from the joint distribution of four or five parameters, due to issues of parameter identifiability. Moreover, obtaining accurate results with this approach requires careful choice of starting point and tuning some other parameters. In general the performance of the algorithm will depend on the initial point. We tested the use of both random starting points and MAP estimators as starting point. The former enables us to detect when several modes are present in the posterior distribution; the latter helps to focus sampling near to the most likely parameter estimate and to quantify uncertainty in it. However, it is also important to note that using MAP estimator as a starting point is not helpful in all cases. More precisely, if the MAP estimator is not a global minimum but a local minimum, then the chain could get stuck around this point. Therefore, it requires careful analysis, comparison and synthesis of the results obtained with these different approaches.

3.5 Methods, Datasets, and Experimental Design

In this section, we describe the datasets that we have in the T2DM and ICU settings, the experiments that we design to present our numerical results, and the methods that we follow to perform parameter estimation and forecasting. Depending on the specifics of each case and to reflect the real-life situation, we designed slightly different experiments in the T2DM and ICU settings. However, the mathematical solution approaches for parameter estimation and forecasting stay the same for both settings because we use similar mechanistic models. In this opening discussion we first describe the features that are common to both the T2DM and the ICU settings. The two following subsections 3.5.1, 3.5.2 then detail features specific to each of the two cases.

Because we use a linear, Gaussian stochastic differential equation to model the BG level, our forecast is a Gaussian characterized by its mean and standard deviation. Hence, rather than having a point estimate for the future BG levels, we obtain a normal random variable for each prediction. In testing predictions of the model it is natural to check if $1-$ and $2-$ stdev intervals around the respective means capture the true BG levels. Note that the probability of a Gaussian random variable to take values within $1-$ and $2-$ stdev regions around its mean are $\sim 68\%$ and $\sim 95\%$, respectively.

We define the observational noise covariance $\Gamma(\theta)$, defined in (3.17), to be a diagonal matrix with form $diag(\Gamma(\theta)) := \lambda * Lm(\theta)$. Whilst we could estimate λ alongside θ, from the data, we have chosen a heuristic to set it in advance. Specifically we found that above a value of around 0.3 all forecasts were very noisy and contained little predictive value; on the other hand, below 0.3 results appeared to be fairly robust to the value chosen for λ; in all the experiments presented in Section 3.6 we choose $\lambda = 0.1$.

3.5.1 T2DM

3.5.1.1 Model, Parameters, and Dataset

In this setting, we use the model (3.8) with the function m_k defined as in (3.9). Hence, there are five parameters to be estimated: basal glucose value, G_b, BG decay rate γ, the measure for the amplitude of BG oscillations, σ, and a and b, which are the parameters implicitly modeling the time needed for the rate of glucose in the nutrition entering the bloodstream to reach its maximum value and the total time needed for this rate to decrease back to 0. We assume that the prior distribution is non-informative and initially the parameters are independent, except for a constraint on the ordering of a and b. We determine realistic lower and upper bound values for each of them, define $\Theta' := [0, 750] \times [0.01, 0.5] \times [0, 100] \times [0.01, 0.05] \times [0.01, 0.05]$ (in the order of $G_b, \gamma, \sigma, a, b$), and then define Θ from Θ' by adding the constraint $a < b$. We thereby form the prior distribution as defined in (3.22). Recall that these bounds define the constraints employed when we define the parameter estimation problem in the optimization setting for the MAP point. The set Θ is determined from clinical and physiological prior knowledge, and by simulating the model (3.6) and requiring realistic BG levels. Data are collected from three different T2DM patients. For each patient the dataset consists of the meal times, the glucose amount in the meal and BG measurements along with the measurement times. More detailed information on the dataset such as number of measurements, recorded meals, and mean glucose value over training, testing or over entire data sets can be found in Table 3.1.

3.5.1.2 Parameter Estimation and Uncertainty Quantification

We perform parameter estimation for three patients separately. First, we estimate parameters by using data over four consecutive, non-overlapping time intervals with optimization and MCMC approaches. Besides estimated values, we also provide UQ results. In the optimization setting, we use the Laplace approximation as discussed at the start of subsection 3.4.3. The optimal parameters determine the mean of the Gaussian approximation, and the inverse of the Hessian matrix becomes the covariance matrix, providing the tools for UQ. In the MCMC approach, we use the resulting random samples for UQ.

3.5.1.3 Forecasting

We adopt a train-test set-up as follows. Since the health conditions of the T2DM patients are unlikely to change over time intervals that are on the order of days, we design an experiment in which we use one week of data for training the patient-specific parameters. Then, we use the estimated parameters to form a patient-specific model and use this model to forecast BG levels for the following three weeks, using the known glucose input through the meals; this leads to a three-week testing phase. From a practical patient-centric point of view this leads to a setting in which forecasting BG levels for the following three weeks requires patients to collect BG data for only one week in every month, and then the patient-specific model will be able to capture their dynamics and provide forecasts based on nutrition intake data over the rest of the month.

3.5.2 ICU

3.5.2.1 Model, Parameters, and Dataset

In the ICU setting, we use the model (3.14), and there are now four parameters to be estimated: basal glucose value, G_b, BG decay rate, γ, the parameter used to quantify the amplitude of the oscillations in the BG level, σ, and a proportionality constant, β to scale the effect of insulin IV on the BG level. Similar to what we did in the T2DM setting, we find realistic lower and upper bounds for the unknown parameter values and set $\Theta := [0, 750] \times [0.02, 0.5] \times [0, 100] \times [20, 110]$ to obtain

Patient ID	patient 1	patient 2	patient 3
Total # glucose measurement	304	211	91
Total # meals recorded	122	76	46
Total # days measured	26.6	27.67	28.12
Mean measured glucose	113±25	127±32	124±26
Training set: # glucose measurement	80	53	29
Training set: # meals recorded	26	18	15
Training set: # days measured	7.02	7	7.05
Training set: mean measured glucose	112±25	116±28	125±24
Testing set: # glucose measurement	224	158	31
Testing set: # meals recorded	96	58	62
Testing set: # days measured	19.58	20.67	21.07
Testing set: mean measured glucose	113±25	130±33	123±27

Table 3.1: Information about the data set that is used in the T2DM setting, which is collected from three different T2DM patients. Note that there is a considerable variability between the data collection behaviour of each patient, which is also reflected in the number of recorded measurements and meals. Also, recall that we intentionally used one week of data for training and the following three week of data for testing.

the prior distribution as defined in (3.22). In this case, we impose two further linear constraints, namely $G_b - 3.5 * \beta < 115$ and $\beta - 1110\gamma < 10$. These constraints are imposed to ensure that the model predictions remaining biophysically plausible, and are determined simply by forward simulation of the SDE model; the resulting inequality constraints do not overly constrain the parameters in that good fits can be found which satisfy these constraints, and yet they yield more realistic BG level behavior than solutions found without them. Thus as in the T2DM case, we the bounds and constraints chosen are based on physiological knowledge and requiring simulated BG levels resulting from values within the region Θ to be realistic.

In this case, the dataset consists of the rate of glucose in the nutrition and the rate of insulin infusion along with the times at which there is a rate change. It also contains the BG measurements and the measurement times. Summary statistics about the data set that is used in the ICU setting can be found in Table 3.2. Note that in this case, we used all available data for each patient to perform parameter estimation and forecasting, and all three ICU patients are non-T2DM.

Patient ID	patient 4	patient 5	patient 6
Total # glucose measurement	177	204	271
Total # days measured	13.99	16.8	24.48
Mean measured glucose	141±18	151±32	151±43
Training set: average # glucose measurement	14.13	13.5	14.07
Testing set: average # glucose measurement	1	1	1

Table 3.2: Information about the dataset that is used in the ICU setting, collected from three ICU patients who are not T2DM. Because of the experiment we designed the training sets are moving with by overlapping with each other. So, we provide average number of glucose measurements over these moving windows. Also, since we forecast until the next measurement time following the training time window, each testing set contains only one glucose measurement. Other information that is included in Table 3.1, but not here, such as mean measured glucose over training set(s) is neither meaningful nor helpful in this setting.

3.5.2.2 Parameter Estimation and Uncertainty Quantification

We use both the optimization and MCMC approaches for parameter estimation in a patient-specific manner, in this setting, too. However, for UQ, we use only MCMC to estimate the posterior mean and variance on the parameter; this is because there were cases where it was not appropriate to use the Laplace approximation, something that will be explained in more detail in Section 3.6.2.

3.5.2.3 Forecasting

Patients in the ICU exhibit BG time-series that are very different from T2DM patients; in particular the time-series is often non-stationary in complex ways and on different time scales. On slower time scales, patients eventually leave the ICU because their health either improves or declines. But there can be fast time scale changes too due to interventions and/or sudden health-related events, such as a stroke. These health changes will lead to changes in the best-fit parameters of the model; in other words the patient-specific model itself may change abruptly, in contrast to the T2DM case where changes in the best-fit parameters typically occurs on a much longer time-scale, and reflects gradual changes in health condition. To avoid compensating for different values of parameters over longer time intervals, and to make more accurate predictions, we use only one day of data for parameter estimation in the ICU. Moreover, to construct an experiment that reflects real-life scenarios, we need be able to estimate the model parameters with smaller size datasets than in the T2DM case, because of the imperative of regular intervention within the ICU setting, typically on a time-scale of hours. As a consequence our train-test set-up in this case differs quantitatively from the T2DM case. The training sets for each patient consist of approximately one day of data over a moving time intervals, with end points chosen to be BG measurement times. Thus, the time windows are obtained by moving the left end point to the next BG measurement time and choosing its right end point with the constraint that it contains approximately one day of data and the new time window is not contained in the previous one. So, in this case, there is a large overlap between the consecutive time windows of the training sets.

On the other hand, because of rapidly changing conditions, forecast of BG levels needs only to be accurate over shorter time-scales, too. Indeed, in general, it is important to know glycemic dynamics on the order of hours (not days) to manage the glycemic response of patients. So, for each training set, the left end point of the time window of the corresponding testing set is chosen to be the right end point of the time window of the patient's training set. Then, we choose the right end point of the test set to be the next BG measurement time. We follow the same procedure over the moving time intervals to the end of the whole dataset for each patient. Note that from a practical point of view, this experiment exhibits a real life situation in which we use only one day of data for parameter estimation and then perform forecasting for the next few hours based on the estimated parameters. Such a set-up would be desirable as a support to glycemic management of these patients.

3.5.3 Evaluation Metrics

Before having a closer look at the numerical results in the next section, let us give the definitions of the statistics that will be used to evaluate and compare the forecasting capability of the models. Let $\{y_i\}_{i=1}^N$ denote the true BG measurements over the predefined testing time window for an experiment. Let $\{\hat{y}_i\}_{i=1}^N$ denote the forecast at the true measurement time points obtained by a model. Note that if the model is a stochastic one, $\{\hat{y}_i\}_{i=1}^N$ represents the mean of the model output, while it is simply the model output for a deterministic ODE model. When a stochastic model is used, it is natural to obtain a confidence interval as this may be obtained as a direct consquence of the fact that the model output is in the form of a random variable; such a model output cannot be obtained for an ODE type of a model when parameters are learned through optimization. However, by using appropriate parameter and state estimation techniques, it may again be possible to obtain a similar kind of confidence interval for the model output which is in the form of a point-estimate. When we have probabilistic forecasts we let $\{\epsilon_i\}_{i=1}^N$ denote the corresponding standard deviation for each forecast at the true measurement points so that we can form 1- and 2-stdev bands as $[\hat{y}_i - \epsilon_i, \hat{y}_i + \epsilon_i]_{i=1}^N$ and $[\hat{y}_i - 2\epsilon_i, \hat{y}_i + 2\epsilon_i]_{i=1}^N$, respectively. Then, for each model, we can compute the percentage of true measurements, $\{y_i\}_{i=1}^N$, that are captured in their respective 1- and 2-stdev bands. These two percentages will be two of the evaluation tools that will be used in evaluation below. In addition, in some cases, we will use standard measures such as mean-squared error (MSE), root-mean-squared error (RMSE) and mean percentage error (MPE), which are computed as follows.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \quad RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \quad MPE = \sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i} * 100.$$

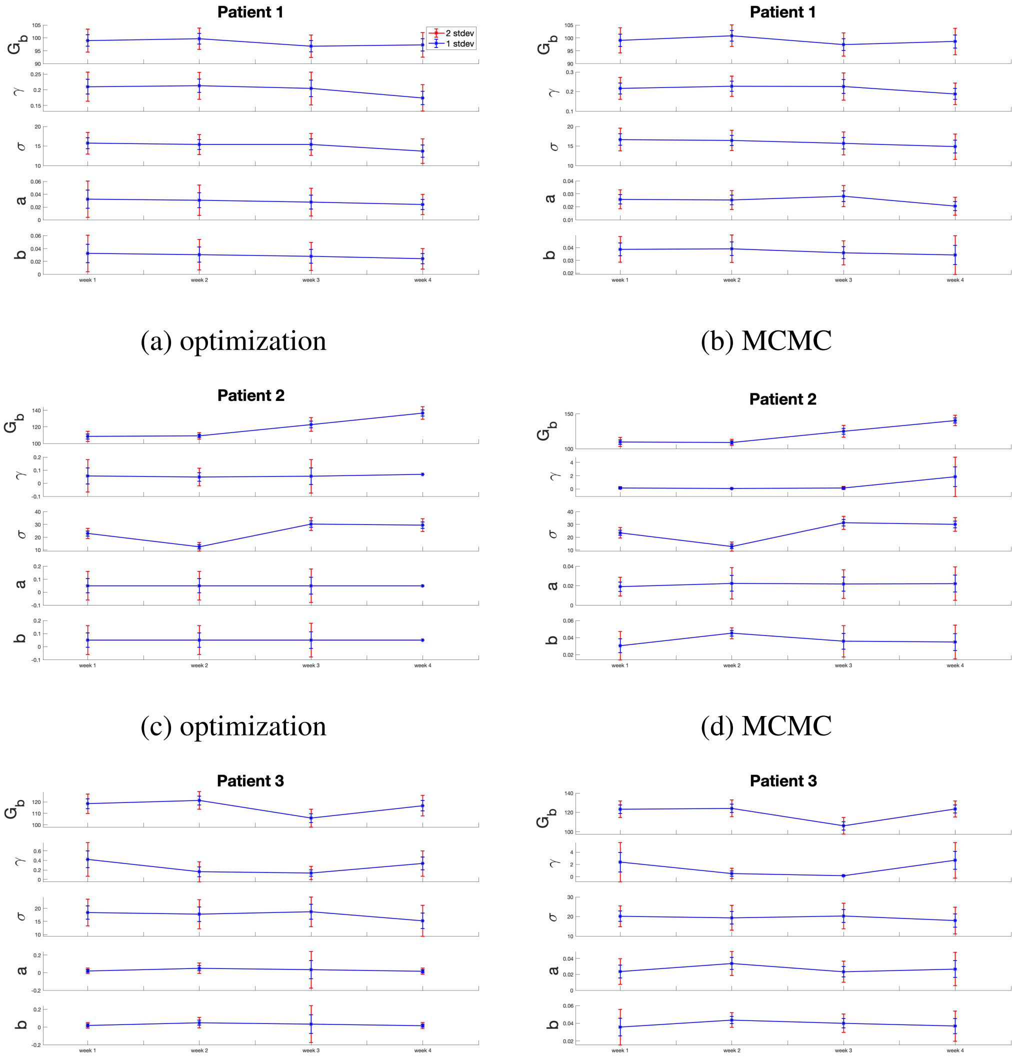
3.6 Numerical Results

In this section we present numerical results concerning the simple, yet interpretable, model introduced in this paper; we refer to this as the minimal stochastic glucose (MSG) model from now on for ease of exposition. The two primary conclusions are that:

- we can achieve good accuracy forecasting future BG levels in both the T2DM and ICU settings, and the uncertainty bands with which we equip our forecasts play an important role in this regard;

- we can learn a substantial amount about the interpretable parameters within the models, with possible clinical uses deriving from the parameter estimates, and from tracking them over time, again using the uncertainty measures that accompany them as measures of confidence.

We justify these conclusions using T2DM self-monitoring data from a previous prospective self-management trial, and using retrospective ICU data extracted from the Columbia University Medical Center Clinical Data Warehouse. The combination of simple predictive model and data acquisition model accounts for the uncontrolled and complex nature of the data, including data sparsity, inaccuracy, noisiness, non-stationarity, and biases resulting from the health care process [5, 13, 185, 183, 184, 186, 235, 322], whilst also being interpretable and leading to patient-specific parameter inference and prediction. To forecast BG for individuals we first solve the parameter estimation problem to entrain the model to the individual, and we present the numerical results in this order. Even though the MSG model is relatively simple physiologically it is not always identifiable, given data. For example, having two parameters, γ and β, related to BG decay rate in the ICU context made it hard to identify these parameters accurately given the sparsity of the data, the non-stationarity of the patient, and the complexity of the glycemic dynamics. Despite lack of identifiability of some parameters, parameters as estimated lead to models which are able to forecast and represent the glucose-insulin dynamics. For example, in both the T2DM and ICU cases, the UQ results along with estimated parameter values indicate that the estimates of both the basal glucose rate, G_b, and the proportionality constant between the basal glucose rate and the variance of the glycemic dynamics, α, reflect realistic values with uncertainties that manage to capture future data but remain narrow enough to potentially delineate different treatment pathways. To answer whether the parameter estimates, forecasts, and uncertainty quantification *are* good enough to impact clinical understand and decision-making or to construct physiologically-anchored phenotypes [6, 7, 8] would require evaluation, e.g., manual chart review in conjunction with a qualitative trial of clinical decision-making or a phenotyping analysis respectively. In the absence of these analyses we will rely on face validity validation [91, 164, 431] of the forecasting and UQ capturing future measures as well as a host of quantitative measures of forecast accuracy. We also reemphasize that the parameter estimates themselves may be useful as they carry information about gradual disease progression (T2DM) and sudden changes in health condition (ICU).



(a) optimization

(b) MCMC

(c) optimization

(d) MCMC

(e) optimization

(f) MCMC

Figure 3.1: Parameter estimation and uncertainty quantification in the T2DM setting. The left-hand panel is obtained with optimization and the right-hand panel is obtained with MCMC, both are in a patient-specific manner. It shows that the point estimates obtained with two approaches are very close to each other in most cases. Also the width of the 1- and 2-stdev intervals, which are obtained with Laplace approximation (in the optimization case) and directly from the approximate posterior samples (in the MCMC) setting, are also agreement with each other. In addition, obtaining estimated values that are in alignment with real physiological values, these results enforces the reliability of the parameter estimation results.

3.6.1 T2DM

We will start by showing numerical results for parameter estimation and forecasting based on the real-world data collected from T2DM patients. These results demonstrate the effectiveness of the MSG model in capturing the patients' BG dynamics. Specifically the effectiveness is reflected in the estimated parameter values and in the efficacy in forecasting future BG levels, using these parameters, over time periods of length up to three weeks.

3.6.1.1 Parameter Estimation

Our numerical results exhibit three substantive pieces of evidence which support the validity of the model and its potential effectiveness for both understanding the physiologic state of an individual, and for forecasting for that individual, in the context of T2DM. *First*, the estimated model parameter values and their evolution over time are physiologically meaningful. That is, the estimated values reflect the patient's state as evaluated given available data. Moreover, the evolution of the estimated parameter values over time reflects changes in the patients' states in a manner consistent with both the data and what is known about the non-stationary nature of T2DM. *Second*, the UQ intervals for the estimated parameters are physiologically plausible and have three features that make the model potentially useful: (i) relative to the value of the estimated parameter, the UQ intervals are wide enough to provide information on the reliability of the point estimates of the model parameters (ii) the UQ intervals evolution over time, demonstrating a sensitivity to time and the ability to adapt to non-stationary patients, and (iii) the UQ intervals are narrow enough to plausibly be used to differentiate behavior choices. And *third*, the UQ and parameter estimation appears to be robust; different estimation methods arrive at similar results. A comparison of the estimated parameter values and corresponding UQ intervals obtained using optimization and MCMC are very similar in almost all of the cases, implying robustness of the estimates and a relative insensitivity to the estimation methodology. Together these features imply that with a reasonable inference scheme this model could potentially provide useful information for clinical decision making and deeper clinical understanding of the patient robustly.

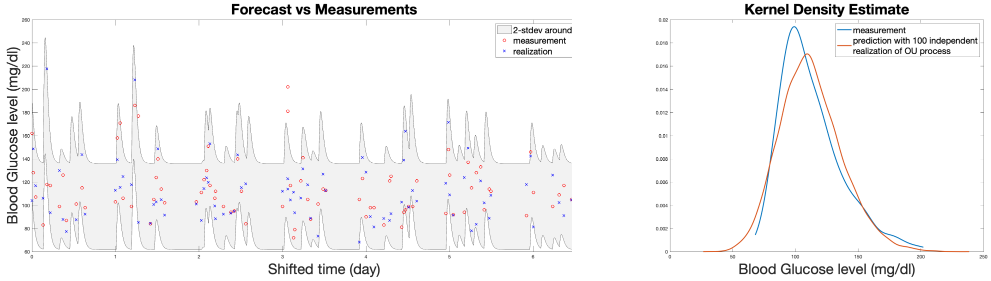
To demonstrate that the estimated parameters are physiologically meaningful, consider Figure 3.1 where we see the point estimates as well as UQ intervals for all parameters and all three patients obtained with optimization and MCMC methods. The estimated basal glucose, G_b, values are in the ranges of $\sim 95 - 105$ mg/dl, $\sim 105 - 140$ mg/dl, and $\sim 105 - 125$ mg/dl over the course of four weeks for patients 1, 2, and 3, respectively. These values are indeed in the expected ranges based on the BG measurements of these patients. In addition, as we will describe later in more detail, the estimated parameters are able to usefully predict the glycemic mean and variance of patients with T2DM on time-scales of around three weeks. The forecast of glycemic mean and variance in response to nutrition is limited to three weeks because the data shows non-stationary effects over longer time-intervals. Figure 3.1 reveals parameter changes that, over four weeks, are significant enough to render predictions less reliable, whilst on a three-week time horizon they are accurate.

To show that the UQ intervals are potentially useful in practice, once again consider Figure 3.1. The range of UQ intervals for each estimated parameter in most of the cases contains physiologically plausible parameter values that are tight enough to enforce the reliability of the point estimates of the parameters. To quantify this statement we computed the coefficient of variation, defined as the standard deviation over the mean. This measure is generally interpreted as a dispersion of the probability density and can be interpreted as the variability of the distribution in relation to the mean. Smaller values of coefficient of variation imply less variability or dispersion within the population and that the distribution is accumulated around the mean. When the coefficient of variability is low, point estimates of the mean are particularly meaningful and represent the population well whereas when the coefficient of variation is large, the mean is less representative of the population as a whole. Note that the population we are quantifying here is not of different patients, but rather a population of different forecasts, parameter estimates, or realizations of the stochastic model, for the same patient at a given time. For basal glucose rate, G_b, and amplitude of oscillations, σ, the coefficient of variation is in the $\sim 2-3\%$ band and $\sim 8-20\%$ band, respectively for all three patients, implying that the mean of the estimated G_b is a very good point estimator. Even though the coefficient of variation values are not as small for σ, these values are still quite small and demonstrate limited dispersion. Together these results support the reliability of the point estimates that are used to form patient-specific models to describe dynamics of each patient. In addition, the evolution of these UQ intervals for each parameter over four weeks, present in Figure 3.1, demonstrates their sensitivity to time and the model's ability to adapt and capture the non-stationarity in the dynamics of patients over time.

We can see the robustness of the estimated parameter values by comparing parameter estimates using two different methods, MCMC and optimization. The results are shown in Figure 3.1; the left and right columns show parameter estimates using optimization and MCMC, respectively. The point estimates as well as the corresponding UQ intervals for basal glucose value, G_b, and amplitude of oscillations, σ, obtained with optimization and MCMC are very close to each other. Some parameters have more variation between methods; specifically, the rate of decay to the basal glucose rate, γ, and the meal function nutrition absorption parameters a and b do show variation between the results obtained with optimization and MCMC methods. This variation does not seem to have substantial effect on the model's ability to represent patient dynamics. The overall result is a model whose ability to represent the data is relatively insensitive to parameter estimation techniques.

3.6.1.2 Forecasting

The stochastic modeling approach is simple in the sense that we have few state and parameters and the model's high-frequency dynamics are represented as a diffusion process whose centroid is governed by processes such as physiology-driven mean reversion. In contrast, this modeling approach is complex because a stochastic process doesn't have an explicit glucose trajectory—a particular glucose value—at a given time but rather is a function that defines a *glycemic distribution* at every time point, e.g., with a mean and a variance. Because of this subtly, the model is both intuitive—it reflects what we know and do not know about glycemic dynamics at given, unmeasured, time—and it is foreign because there is not an explicit glycemic trajectory. However, we can construct an *example glycemic trajectory*, or a realization



(a) An arbitrary realization with BG measurements

(b) Kernel density estimates

Figure 3.2: In (3.2a) a realization of the estimated Ornstein-Uhlenbeck process is shown over the first week of the test data along with the true BG measurements, and in (3.2b) kernel density estimates of BG measurements and realizations of the estimated model are shown. In 3.2a, the red circles show true BG measurements, the blue crosses show an arbitrary realization of the model output, and the gray area represents the estimated 2-stdev band around the mean of the model output. This figure shows rationale behind the MSG model. Comparison of the true BG measurements and an arbitrary realization of the estimated distribution implies that they could indeed be considered as two different realizations of the same random process, as most of the true and simulated BG levels stay within the 2-stdev band.

of the stochastic process by sampling the SDE-defined glycemic distribution at every time point. In another words, the realization is one of the infinitely many possible trajectories that the stochastic process could follow when it is realized. Similarly, we assume the collected measurements represent a realization of a random process that is described by the solution of our SDE model. Together these pieces form a framework within which we interpret and evaluate the model. As such, we evaluate the model along two pathways, a face validity pathway that is mostly motivated by potential clinical decision-making, and a more statistical-based pathway that is motivated by our desire to be quantitative. In a sense, both evaluations address whether the data could plausibly be generated by the model.

The *first* evaluation—face validity—is to consider whether the model can capture the dynamics qualitatively. Because the model's forecast is in the form of a distribution, the forecast we have to evaluate is anchored to the mean and standard deviation. An initial inspection of Figure 3.2a where the red circles represent the true BG measurements and blue crosses represent one *realization* of the model over only the first week of the test set does indeed seem to represent the data well.

The *second* evaluation quantifies how plausible it is that the data we observe could have originated from the model. We quantify this plausibility using the two-sample Kolmogorov-Smirnov (KS) test. To start, Figure 3.2b shows the kernel density estimates obtained from the BG measurements (blue curve) and from ensemble of 100 different realizations of the estimated stochastic process (red curve) over the same time period in Figure 3.2a. To generate the data to compute the model-based density estimate we select a sample from the distribution defined by the model at every time point. The model can also be visualized as a probability density function; for example, the kernel density estimate of a model realization is the probability density function of the distribution and the real data are shown in the right panel of Figure 3.2. The similarity of the behavior of the *estimated* realization (blue crosses) with the *assumed* realization (red circles) of the stochastic process in Figure 3.2a and kernel density estimates in Figure 3.2b support the idea that both are plausible draws from the same distribution. To calculate the two-sample Kolmogorov-Smirnov (KS) test we created data sets: (i) resampled 1,000 points from the raw BG measurement data, and (ii) a realization of the estimated stochastic process over the same time period as shown in Figure 3.2a. The two-sample Kolmogorov-Smirnov test did not reject the hypothesis that the two samples came from the same distribution with a p-value of $9.8217 * 10^{-9}$. This implies that our initial assumption, which is that the

BG values can be described by our simplified stochastic model, is indeed a valid assumption in this setting.

Given this understanding of the model, our evaluation of BG forecasting focuses on evaluating the ability of the model to estimate and track the mean and variance of BG levels. The MSG model's forecast is in the form of a distribution because it is stochastic and is therefore represents the glycemic distribution at every time point. In this way, the only forecast we have to evaluate is the mean and variance. And, clinical understanding and decision-making is done relative to the mean and variance of glycemic dynamics. As such we have two key results. *First*, the forecasted mean of the MSG model output, the stochastic model for BG movements, captures the essence of the behavior of true BG measurements in a realistic way. And *second,* the forecast uncertainty as quantified using the standard deviation of the process encapsulates a large percent—94% on average over the three patients—of true *future* BG measurements while remaining narrow enough to delineate changes in nutrition input and potentially treatment strategy. Surprisingly, the forecast uncertainty is *more narrow than the empirical uncertainty while capturing more of the data*, meaning that the forecast uncertainty captures the future uncertainty of the data more accurately—more narrow but more specific—than the data themselves capture their own uncertainty. This is of course possible because the model is modeling glycemic response, not just the glycemic time-series. Because the optimization and MCMC approaches produced very similar parameter estimation and hence forecasting results, we only used the MAP estimators obtained with the first week of data to form the patient-specific model for forecasting over the following three weeks.

The mean of our glucose model represents the mean glycemic homeostasis and the mean glycemic response to nutrition. Figure 3.3 demonstrates how this mean reflects

	1-stdev %	2-stdev %	model stdev	data stdev
Patient 1	73.66	94.20	19.3700	24.4629
Patient 2	62.66	89.87	25.6625	32.5589
Patient 3	51.61	96.77	22.2388	27.1433

Table 3.3: Percentages of the true BG measurements included in the forecasted 1- and 2-stdev bands, in the T2DM setting. Besides visuals provided for forecasting results, this table shows indeed a large amount of true BG measurements are captured in the forecasted confidence intervals. Comparison of the model stdev with the raw BG data stdev shows that the confidence intervals act really as tight bands around the measurement values, hence providing true information about their variance.

the dynamics of true BG measurements. In this figure we estimated the model parameters using one week of data, producing a model of the glycemic homeostasis and response given nutrition input. To evaluate the forecasting ability of this model we then use this model to forecast glucose for the following three weeks. The subfigures of Figure 3.3 show the resulting forecast of BG for patients 1, 2, and 3, respectively for the three weeks after the model was estimated. In each subfigure, red circles represent the BG measurements, the blue curve shows the mean of the MSG model output and the gray area is the 2-stdev band around the mean. We see that the blue curve—the proper forecast—encapsulates the behavior of the true measurements for each three patients.

Not all glycemic responses follow the mean, and forecasts carry uncertainty. One particularly important and challenging task of a forecast is accurate estimate of uncertainty. Because of the nature of our stochastic model, uncertainty is quantified naturally using the standard deviation of the model process. Figure 3.3 demonstrate the effectiveness of the models' ability to capture relevant forecast uncertainty with two standard deviation (2-stdev) bands around the model mean; these bands capture nearly every future BG measurement. These results are further quantified in Table 3.3 that shows summary statistics for how often the measurements were captured by the 2-stdev bands as well as the estimated standard deviation of BG measurements and the empirical standard deviation obtained directly from the raw BG measurement data. Being able to contain $\sim 90 - 97\%$ of the true BG measurements in these confidence regions with a smaller model standard deviation than the empirical standard deviation for all three patients is an indicator of this model's capability in capturing the patient dynamics and hence its predictive capability. This model is providing substantial forecasting information beyond what is available given the data alone.

3.6.1.3 Comparison of Forecasting Accuracy with LDP Model

In this section, we will compare the forecasting accuracy of the T2DM version of the MSG model with a well-known model developed by Ha & Sherman [159]. It is important to note that this model, the longitudinal diabetes pathogenesis (LDP) model, was designed to understand diabetes progression, not for forecasting future BG level purposes. The experiment in this setting will be the same as described above in Section 3.6.1.2. This model consists of a set of coupled ODEs. To estimate the unknown model parameters within the LDP model we use a constrained ensemble Kalman filter (EnKF) algorithm, whose details can be found in [4]. As a result of using this type of an algorithm, whose output comprises an *ensemble* of estimates, we can also assess the uncertainty in the estimated parameters, using the ensembles. Moreover, we can propagate the uncertainty in the estimated model parameters to quantify the uncertainty in the forecasted model states. Thus, using this approach, we can obtain confidence bands, as we do with the MSG model. The LDP model that we use for comparison has 11 model parameters. However, we will not attempt to estimate all these parameters as it is not a feasible task to achive with the sparse data that we have available. Instead we set some model parameters to reasonable values based on the literature, and estimate the remaining ones using the constrained EnKF method. More precisely, we perform the same experiment by estimating three different sets of parameters, $\{\sigma, SI\}$, $\{\sigma, SI, hepaSI\}$, $\{\sigma, SI, hepaSI, r20\}$, and setting the remaining parameters at known default values. The comparison results can be found in Table 3.4.

The results in Table 3.4 show that the MSG model is better at forecasting future BG levels in T2DM patients than all variants of the LDP model considered. *First*, even though we fit the normal distributed model output to the data rather than directly fitting the mean of the model output to the data, we can achieve smaller MSE and MPE than all different variations of the LDP model for all three patients. *Second*, we see the advantage of using a stochastic model which quantifiies the level of certainty in the BG predictions for both the LDP and MSG cases. It is worth noting that the MSG model is based on learning parameters of a stochastic model, whilst the LDP quantifies uncertainties by learning an ensemble of parameters; this may contribute to the differences between them at the level of uncertainty prediction. The pecentages in Table 3.4 show that the MSG model is substantially better in capturing the true measurements in the corresponding confidence bands. Note also that the MSE is also smaller for the MSG than for the LDP, demonstrating that it is preferable as a point estimator, as well as probabilistically. In summary the MSG model is preferable to the LDP for decision making in the context we use here, as it gives a better point forecasts and better confidence bands, enabling knowledge of possible high and low values for future BG levels.

Finally, we want to note also that the datasets belonging to Patient 1 and Patient 2 here were also used in [10] to compare the efficacy of some other data assimilation techniques to forecast future BG levels. Among different filtering approaches and mechanistic models, the best performance was achieved by a modified dual unscented Kalman filter (UKF) that estimates both the states and unknown model parameters that is used along with the Ultradian model. With this approach, the MSE for patient 1 is reported to be 680 $(\text{mg/dl})^2$ whereas it is 950 $(\text{mg/dl})^2$ for patient 2. This shows that we obtain better forecasting accuracy with the MSG model for patient 1, however, the forecasting accuracy is better for patient 2 with the UKF approach used with the Ultradian model. In addition, the authors used the well known Meal model, introduced in [98] for forecasting future BG levels. For this model, among various different filtering techniques, they achieved the best accuracy again with UKF. The MSEs obtained with the Meal model along with UKF are 730 $(\text{mg/dl})^2$ and 1300 $(\text{mg/dl})^2$ for patients 1 and 2, respectively. The MSG model achieves better forecasting accuracy than the Meal model for both of the patients. These comparisons are another indicator that the MSG model can provide accurate forecasting results for T2DM patients without including exogenous insulin as a state variable and with a relatively simpler representation of the underlying physiology.

Patient 1						
		1-std %	2-std %	mse	rmse	mpe
MSG Model		73.66	94.20	403.94	20.10	12.84
LDP Model	σ, SI	41.96	65.62	524.03	22.89	13.72
	σ, SI, hepaSI	40.18	66.96	483.99	22.00	13.77
	σ, SI, hepaSI, r20	43.30	65.62	487.89	22.09	13.59
Patient 2						
		1-std %	2-std %	mse	rmse	mpe
MSG Model		62.66	89.87	1123.40	33.52	17.35
LDP Model	σ, SI	20.89	37.34	1563.20	39.54	21.00
	σ, SI, hepaSI	15.82	32.91	1952.30	44.18	24.20
	σ, SI, hepaSI, r20	18.35	33.54	1630.60	40.38	21.71
Patient 3						
		1-std %	2-std %	mse	rmse	mpe
MSG Model		51.61	96.77	586.32	24.21	17.11
LDP Model	σ, SI	29.03	50.00	1043.40	32.20	18.98
	σ, SI, hepaSI	30.65	53.23	1080.90	32.88	18.69
	σ, SI, hepaSI, r20	19.35	46.77	1117.60	33.43	19.81

Table 3.4: Comparison of the forecasting results with two different models. For each different case of the LDP model the results in the corresponding row shows which parameters are estimated during the whole forecasting experiment. Note that we obtain better forecasting accuracy with the MSG model than with the LDP model. Furthermore, for the LDP model, the forecasting accuracy decreases as the number of parameters being estimated increases.

3.6.2 ICU

We now move from evaluating the model with T2DM self-management data to the more complex and difficult case of modeling and forecasting glycemic dynamics in the ICU, where non-stationarity is manifest on much shorter time-scales. Parameter estimation and prediction are, in general, harder in the ICU context because patients within the ICU typically have much more volatile physiological dynamics for at least three reasons: glycemic dynamics under continuous feeding are oscillatory, the patients are acutely ill and their health state changes quickly because of their disease state, and the patients are constantly being intervened on to help them heal. To paint a picture, 90%+ of the patients will not require insulin outside of the ICU but do during their ICU stay, and around 20% of patients have a hypoglycemic episode that would not occur when they are not acutely ill. The ICU is a much more complex forecasting and modeling setting.

3.6.2.1 Parameter Estimation

The difficulties presented in the ICU setting are reflected in our parameter estimation results. Despite these complexities, Our numerical results exhibit four substantive pieces of evidence which support the validity of the model and its potential effectiveness for both understanding the physiological state of an individual, and for forecasting for that individual, in the context of ICU patients. *First*, the model captures the dynamics as meaningfully as possible based on the data. That is, the estimated model parameters are physiologically plausible and represent the observable dynamics. *Second*, the estimated model parameters, which have the most influence in resolving the mean and variance of the BG level, are physiologically meaningful in most of the cases, as was the case in the T2DM setting. *Third*, the changes in the parameter estimation results over moving time windows are realistic and reflective of the expected non-stationary behavior of ICU patients. And *fourth*, the UQ results show that the parameters (basal glucose rate, G_b and the model standard deviation, σ), which have the most influence in resolving mean and variance of BG levels are estimated with more certainty. Having tighter bands around the point estimates for these parameters indicates the robustness of the estimation.

Before we begin the evaluation, first consider Figure 3.4, which demonstrates both the model's relative robustness and its capability of capturing the dynamics and various complexities encountered in different conditions in the ICU setting. These figures show simulated BG values for patient 4 (see Table 3.2) over different training time windows, whose data, nutrition rate and BG measurements, are used to estimate the corresponding model parameters using the optimization approach. Here the the red curves represent the mean of the blood glucose dynamics that are assumed to be oscillatory, the amplitude of oscillations are expected to lie in the gray region as it is the 2-stdev band around the mean, and the BG measurements are shown as red circles and the blue curve shows the tube-nutrition input rate. For simplicity we are considering patient 4 who did not need external insulin, so the tube-feed nutrition is the only driver of the BG level. Each subfigure of Figure 3.4 shows a different training time window that is representative of different circumstances relative to our ability to estimate the basal glucose rate G_b, the decay of glucose to the basal rate, γ, and the parameter that resembles the width of the glycemic dynamics, σ. Figure 3.4a shows a situation where the BG measurements reflect the nutrition rate quite well. In this case all the estimated model parameters are physiologically meaningful and the resulting simulation is a good representative of the dynamics, as can be seen by the parameter and state estimates tracking one another. In contrast, Figures 3.4b and 3.4c demonstrate a situation where the BG measurements do not reflect the nutrition rate over the time window; this failure is seen by the lack of consistency in the movement of the parameters to one another and the nutrition rate. This failure can have one of two sources. First, if there is no change in the nutrition rate over the training time window, it is impossible to estimate the glycemic decay rate parameter, γ. Second, when changes in the BG measurements are uncorrelated with the changes in the nutrition rate, potentially due to changes in health states or other interventions, e.g., other hormone drips, it is also impossible for the model parameters to accurately reflect the physiology as they are accounting for dynamical glucose features they were not designed to accommodate. These issues do not mean the model cannot represent and forecast the glycemic dynamics, it still is usually able to represent glycemic dynamics, but *some of the parameters* might lose their intended meaning. For example, in the two respective examples, despite parameter estimate issues, in both of these cases the estimated basal glucose rate, G_b, and the the parameter, σ that is a measure for the amplitude of the BG level oscillations *are physiologically meaningful* and these parameters are enough to capture the mean and variance of the BG measurements accurately. Moreover, estimated decay parameter, γ, takes an arbitrary value larger than a pre-set threshold resulting in that the mean of the model being estimated as flat. These examples are not the only cases where we observe parameter estimates that are not physiologically meaningful while at the same time the glucose forecast and modeling itself remains accurate. The other examples are all variations on the same theme; we do not have the available data to estimate a parameter accurately, or the data are behaving in a more complex manner, and in both cases, the parameters make up for these data-driven and model-driven short-comings by deviating from their normal roles to render a robust glucose forecast. It is likely that problems such as these will not be eliminated by using more complex data sets and more complex models, because full representation of the relevant processes is out of reach in such non-stationary ICU settings.

With the complexity of ICU data in mind, Figures 3.5 and 3.6 show the time course of parameter estimates for each ICU patient obtained with MCMC and optimization approaches respectively, and demonstrate how the estimated parameters are physiologically plausible. In Figure 3.5 we show estimated basal glucose rate, G_b, the decay parameter, γ, the parameter used as a measure for the amplitude of BG level oscillations, σ, and the proportionality constant, β, for each of the patients. The mean of each parameter, as estimated using MCMC, is shown using blue stars; the parameters are estimated for every forecasting process using data from the previous day allow us to update the model to forecast the glycemic response and states, implying a moving time window of parameter estimates. These parameter estimates are physiologically plausible for all three patients except in a small number of cases. For example, estimates of the basal glucose rate, G_b, were around $\sim 110 - 150$mg/dl, $\sim 140 - 200$mg/dl, $\sim 120 - 200$mg/dl, for patients 4, 5, and 6, respectively, all plausible values given the patient's data. As was the case for the example discussed in the first paragraph of this section, it was not possible to compute good estimates for parameters γ and β in some of the cases.

If we estimate the parameters using optimization—changing the paradigm under which we estimate parameters—we can gain further insight into complexities regarding parameter estimation. Figure 3.6 demonstrates some variability and occasionally unrealistic estimates for basal glucose rate, G_b. In Figure 3.6 we observe times where the basal glucose rate, G_b being estimated very low, too low to be plausible. The reason why the basal rate is estimated incorrectly, however, is not so complex and is in fact correctable. The time periods where the basal rate is incorrectly estimated coincide with time intervals where we cannot estimate the decay rate, γ; this problem occurs again when the nutrition rate is not changed over the course of the training period, making it impossible to estimate glycemic response to nutrition. If, over the course of the optimization, the decay rate is estimated to be too high, it negates the effect of nutrition to the BG rate. Because the model now has much less dependence on nutritional input, it makes up for this by estimating the basal rate, G_b as being higher than it should be. In contrast, if the estimated decay rate, γ, is underestimated then the influence of nutrition on BG is excessive and, to make up for this, the basal rate is underestimated. In this situation we can still calculate the basal rate accurately by estimating the *shifted basal rate*, which is the sum of estimated basal glucose value and estimated effect of nutrition rate. This shifted basal rate is how the model is modeling the glucose in the system, and the calculation for the shifted basal rate is effectively deconvolving how the model is coping with the data insufficiency. This example demonstrates some of the ICU-specific complexity and that, despite the identifiability failure due to data sparsity, the model was robust enough able to estimate the data. And, because the model is relatively simple, this further demonstrates how we are able to pull apart the modeling inaccuracies such that we can understand and account or otherwise compensate for these model errors. In addition, the estimated σ values which are the measures of the amplitude of BG level oscillations, attain physiologically meaningful values, using the optimization approach, as well. This is important because it is generally the amplitude of oscillations that will have the largest impact on clinical decision-making. And finally, despite these difficulties, the BG dynamics were still quite accurate as we will see in the forecasting evaluation (cf Figures 3.7 and 3.8).

Figure 3.5 also shows that the time evolution of the estimated parameters is realistic within the ICU context. In ICU the training time windows move in positive (increasing time) direction increments of measurements—given a measurement the model is estimated using the previous 24 hours of data, $\sim$ 14 data points to forecast the future measurement whenever it comes—so that the consecutive time windows have an overlap of 20-23 hours. This means that the model varies relatively continuously between consecutive time windows. This relative continuity is reflected in Figure 3.5 that shows the time evolution of estimated parameters for all three patients. The choice of the time window used to estimate the model faces the same problem that all moving window approaches: short time windows imply less data and higher estimation variance and long time windows imply poor adaptability in non-stationary settings but have lower variance and ample data. This is an optimization problem we will not tackle here. Instead, we set the window size on the assumption that the patient health state defined by the parameters would not change too much over the previous 24 hours, and assumption that is usually but not always correct. Even though the health condition of the ICU patients can change rapidly, the estimated parameters do not change wildly (in most of the cases), reflecting the expectation under these settings. Nevertheless, the patients are clearly non-stationary and the observed evolution of the parameter estimates, e.g., of the basal glucose value, $G_b$and decay rate, γ, shown in Figure 3.4 reflect this non-stationarity.

And finally, as was the case in the T2DM setting, the model is relatively robust to the methods used to estimate it; however, as can be seen in Figure 3.8 and inferred from the discussion above about parameters and their face validity to physiology, the ICU formulation of the model can have more complex parameter estimation issues compared to the T2DM setting. In particular, in the ICU setting there are some cases where the Laplace approximation does not work well because the parameter misfit solution surface is flat in some parameter directions – a reflection of identifiability issues. In these cases we used MCMC to provide UQ results. In general we observe that the basal glucose rate G_b and the parameter related to variance, σ, both allow for more robust estimation compared to the estimation of γ and β. The robustness of the estimation of σ is important for clinical applications because the variance, σ, is what is used for deciding insulin doses. As a demonstration of the robustness of σ, consider Figure 3.5. Here we can see the 2-stdev band around the mean for σ is tighter than or as tight as the 2-stdev bands for G_b, γ, and β for all three patients. This implies that the MSG model is able to robustly estimate the amplitude of the BG level fluctuations, which again is important to clinicians. On the other hand, considering the plots for γ and β estimation, the width of the 2-stdev bands shows that we are less certain about the estimated values. Remember that both of these parameters are related to the glucose removal rate from the blood. This is, perhaps, an indicator of an identifiability issue for these parameters. But it is also true that we are indeed less certain about this physiology; glucose can be removed at different rates by different physiological processes, e.g., liver versus adipose tissue, and we are not resolving these physiological subsystems. Moreover, due to the non-stationary and sparse nature of the data in the ICU setting, it is harder to estimate some of the model parameters accurately. Separating these inference issues is not possible given the data presently collected in these settings. Nevertheless, the parameters that play a key role in resolving the mean and variance of the BG dynamics can be estimated accurately up to the desired level.

3.6.2.2 Forecasting

Forecasting results in the ICU setting are indicative of two major features of this model: (i) we can capture the trend of BG measurements through the mean of the model and (ii) we can estimate the variance of the BG measurements accurately. Once again, since resolving mean and variance of BG dynamics is central to glycemic management, these results show potential usefulness of this model in the ICU context.

	1-stdev %		2-stdev %		average model stdev		data stdev
	optimization	mcmc	optimization	mcmc	optimization	mcmc	
Patient 4	59.06	64.33	94.15	93.57	16.69	17.48	17.21
Patient 5	60.62	67.36	84.46	87.05	25.70	29.09	29.15
Patient 6	55.81	62.40	86.05	89.54	33.64	37.79	38.12

Table 3.5: Percentages of the true BG measurements included in the forecasted 1- and 2-stdev bands, in the ICU setting. These percentages show that a large number of forecasted confidence intervals include the true BG measurements. MCMC approach provides slightly better rates, which is in accordance with the higher average model stdev in the MCMC case. The average model stdev values obtained with optimization and MCMC for all three patients are smaller than the raw data stdev in all but one case. Together with the percentage values, this means that the confidence bands are tight enough to provide accurate information on the variance of BG levels in the ICU context, as well.

Figures 3.7 and 3.8 demonstrate that the forecasted mean of the model and reflect it encapsulates the essence of the behavior of BG measurements for all three patients. In each of the plots in Figures 3.7 and 3.8, the red circles show the BG measurements and the blue stars are the mean of the model, the gray region is the 2-stdev band around the mean once again obtained separately for each forecasting process with the corresponding patient-specific model. In addition to representing the trend of the BG measurements, the forecasted mean of the model is nearly identical when computed using two independent methods, reinforcing the point that the model is reliable.

To observe the effectiveness of this model in estimating the variance of the BG measurements accurately, consider Figures 3.7 and 3.8 and Table 3.5. Figures 3.7 and 3.8 shows the ability of the models to estimate the variance in glycemic dynamics visually where a large number of true BG measurements are contained in the gray regions that represent the *forecasted* 2-stdev bands around the forecasted mean. These results are quantified in Table 3.5 which contains summary statistics both for optimization and MCMC methods. We see that with one exception, MCMC model estimation for patient 4, *the average model standard deviation is smaller than the empirical standard deviation of the BG measurements, yet the proportion of the BG measurements captured in these regions are in the range of* $84-94\%$ for all three patients with two different methods. These results demonstrate the forecasting accuracy of the MSG model, and imply potential use in the ICU for glycemic management.

3.6.2.3 Comparison of Forecasting Accuracy With The ICU Minimal Model

In this section, we will use the ICU Minimal Model (ICU-MM) introduced in [414] and [173] for the comparison of the forecasting result we obtain with the ICU version of the MSG model. The ICU-MM has twelve unknown model parameters. One of those model parameters is used for the purpose of having units equal on both sides of the equation and set to be 1. Two of the model parameters represent the volume of glucose and insulin distribution space and are set to nominal values from the literature. This leaves us with nine unknown model parameters to be estimated. To estimate these parameters, we use the constrained EnKF method as we did in when fitting the LDP model in the T2DM setting. Recall that with this type of approach, the ensemble enables us to obtain confidence bands for our forecasting results.

patient 4					
	1-std %	2-std %	mse	rmse	mpe
MSG Model	60.23	93.57	343.31	18.53	11.03
ICU-MM	49.71	80.70	335.15	18.31	10.48
patient 5					
	1-std %	2-std %	mse	rmse	mpe
MSG Model	60.62	84.46	1104.20	33.23	19.20
ICU-MM	23.83	49.22	1480.40	38.48	20.52
patient 6					
	1-std %	2-std %	mse	rmse	mpe
MSG Model	55.81	86.05	1927.50	43.90	25.84
ICU-MM	26.74	46.90	2018.40	44.93	25.28

Table 3.6: Comparison of the forecasting results obtained with the MSG model and the ICU-MM. The percentages of 1- and 2-stdev bands that capture the true BG measurements with the MSG model is substantially higher than the ICU-MM. On the other hand, MSE and MPE values are much closer yet the MSG model still provides smaller value for these measures, as well.

The numerical results for the comparison are shown in Table 3.6. *First*, similar to the results in T2DM setting, the percentages of the true BG measurements that are captured in the 2-stdev bands with the MSG model are higher than the ones obtained for the ICU-MM by using the constrained EnKF algorithm. *Second*, the point estimators in the MSG case exhibit comparable, or improved, accuracy in comparison to the ICU-MM: MSE and MPE are also smaller for the MSG model except for patient 4. These results show that with a relatively simple model, we are able to reach the same, or better, accuracy in forecasting BG behavior than a more physiologically based high-fidelity model, with a larger number of unknown model parameters. *Third*, the confidence bands that we use to quantify possible high and low values of BG level could provide better results. The improved accuracy of the MSG model in terms of uncerainty forecasting may be related, in part, to the fact that the model we use is inherently stochastic, and fits the stochastic fluctuations to data; in contrast in the ICU-MM provides uncertainty bands only through the ensemble of solutions which are a product of the algorithm used to fit the data, and not inherent to the model itself. Once again, this improved forecast and uncertainty accuracy is indeed a crucial tool in making decisions regarding future BG levels of ICU patients.

3.7 Conclusion

Summary of the modeling framework: In this paper, we introduce a new mathematical model that describes the glucose-insulin regulatory system in humans. The model was developed with five goals in mind: *(i)* to create a model anchored to real clinical data, and that given these data the model would be useful for personalized parameter estimation and state forecasting [380]; *(ii)* to create a model that was interpretable in the sense that patient specific parameters may be used to explain, and quantify, basic physiological mechanisms; *(iii)* a model which is physiologically simple, even if it was functionally complex, to avoid parameter identifiability problems present in many existing physiological models; *(iv)* a model framework generalizable and adaptable to several contexts including T2DM and glycemic management in the ICU; and *(v)* a model that was amenable to a model-based control environment.

With these goals driving the model development, the model we developed follows a somewhat different approach compared to many other glucose-insulin modeling efforts where the goals of increasing physiological fidelity, or explaining a new physiological subsystem, were drivers. For example, where as others, e.g., Sturis et al. [399] or Lui et al. [259], work to understand and resolve the nature of the fast time-scale oscillations, the model developed here incorporates these sub-day glucose fluctuations into the noise process and the parameter estimation is aimed at capturing the slower moving dynamical properties such as the evolution of the rates of glucose use and production; this is done whilst keeping their compartmentalization, and thus number of parameters, to a minimum. We do this because in many cases we do not have data to support resolution of higher-fidelity physiological processes [184] as is the case in many common real-world data collection settings. And, since our overarching goal and model validation and evaluation metrics are based on the models' ability to forecast future BG levels accurately, for the sake of computational efficiency, we end up developing a lower-fidelity model which is simple yet interpretable and anchored to physiology.

Summary of key results: The model developed here is flexible enough to enable a priori plausible models valid for T2DM and ICU settings. Experiments with T2DM and ICU data demonstrate that this a priori plausibility is borne out a posteriori. The model has physiologically interpretable parameters, which can be estimated robustly based on real-world data. Moreover, the estimated parameter values are physiologically plausible for both the T2DM and ICU settings. Hence, the new model has demonstrable capability to capture the BG dynamics of T2DM and ICU patients; in particular it does so well enough to resolve the mean and variance of their BG levels in both retrospective and predictive modes. This feature of the model reveals its potential for use in glycemic management. It also reveals the potential for future BG level forecasting. After being trained based on one week of data, it can accurately forecast future BG levels for the following three weeks in the T2DM context. On the other hand, in the ICU context, it is capable of capturing the dynamics based on one day of data. Then, it can be used for forecasting BG levels for the following 2-4 hours. In both settings, the choice of mathematical model naturally provides confidence bands for the future forecasting of BG levels. These confidence bands are extremely helpful to have an understanding about how low and how high BG levels could be in the future, and hence for the design of glucose or insulin uptake strategies to ameliorate undesirable health effects.

Model development constrained by real world data: Restricting model development to the constraints imposed by readily available real world data is a severe, but important, restriction. We can hypothesize how physiology might work in detail, and we can envisage experiments to gather new datasets that could exist to test our hypotheses; but we have not yet exploited data that are readily available to forge an understanding of what can be explained and predicted given current data acquisition instruments, cost constraints on data acquisition and time-constraints required for real-time prediction. To help facilitate the circular process of allowing our knowledge of systems physiology to inform and impact how people and clinicians manage the health of people, and help allow the gaps in understanding at the bedside to help us choose impactful **systems physiological problems** to focus our efforts on, we need a bridge between these worlds, and the bridge proposed here is through inference with data based on simple yet interpretable models.

Application in clinical settings: Within a clinical setting there are two scenarios where model-based efforts could be of potential help: (i) obtaining deeper understanding of the patient-specific attributes of the glucose-insulin regulatory system; this requires accurate parameter estimation; (ii) guidance for immediate decision-making such as insulin administration and glycemic management; this is a situation where we can tolerate some inaccuracy with parameter estimation provided that the state forecasts, including the uncertainty bands, are accurate and robust. In the context of the model introduced here we have shown situations where the model parameter estimates are accurate as well as situations where model parameter estimates are not accurate. Nevertheless, in the situations where the parameter estimates are not accurate, the state forecast accuracy remains robust, and the parameter estimate failures can be explained and for a large part mitigated. For example, in some circumstances in the ICU we cannot directly estimate the basal glucose rate accurately, but we are nevertheless able to obtain an accurate estimate of the rate at which glucose returns to its base value. In many situations we demonstrate that this is enough to make accurate short-term forecasts; and also provides a starting point for more fundamental physiologically-based systems. A key requirement when translating the model framework to a clinical setting is quantification of uncertainty in predictions. In this context our modeling effort was a success in both T2DM and ICU settings.

Blood glucose forecasting summary: The MSG model works well at estimating and forecasting blood glucose mean and variation boundaries in T2DM and ICU settings. For example, the model-based forecasts have more forecasting accuracy while retaining tighter uncertainty bands compared to measures derived from the data alone. The model identifies different characteristic behaviors between T2DM and ICU patients, demonstrating both generalizability and robustness of the models with respect to forecasting. Moreover, in these two scenarios the models are able to cope with the relative pace of non-stationarity of the patients, order weeks and order days for the T2DM and ICU settings, respectively. This demonstrates both the efficiency of the MSG model and its flexibility. Given these results and the fact that the model is simple and interpretable with understandable parameters implies a potential for providing a new perspective in understanding the glucose-insulin system in humans.

Comparison of the efficacy of the model for T2DM and ICU settings: The T2DM and ICU contexts are very different settings, primarily because of the time-scales on which parameters change, and the different relative importance of external events not included in the model; this difference imposes different needs in the two settings. For example, the change of the health states of the T2DM patients are in the order of days, or even weeks, whereas health states change on the order of hours for ICU patients. Keeping these case-specific differences in mind, one obvious way to compare the effectiveness of the model in these two settings is through the forecasting results. However, unlike other fields such as atmospheric physics, biomedicine is mostly missing a standardized and normalized techniques for context-sensitive forecast verification and evaluation, especially in regard to clinical effectiveness of the forecast. Because of this gap, evaluation of the models and quantitative comparison of their potential usefulness in a context-dependent way is not possible. Regardless, it is important to emphasize that we do not expect the results for the two settings to have the same accuracy due to the characteristic differences mentioned before. More precisely, comparison of Figures 3.3, 3.7, and 3.8 shows that the mean of the model output in ICU setting does not look as close to the true BG measurements as the same comparative forecasts in the T2DM case. Once again, this situation is expected due to highly non-stationary behavior of ICU patients. However, in this setting, it could be argued that being able to forecast the variance of glycemic dynamics to identify, e.g., hypoglycemia, could be more important to capture than the dynamics with the mean model output. These different needs are context-specific, and without context-specific evaluation machinery, direct, quantitative comparisons are not yet possible. Instead we are left showing Figures 3.7c and 3.8c that demonstrate the model can forecast a hypoglycemia event on day ~ 13 for patient 6, a feature that does demonstrate the context-specific effectiveness of the model in the ICU setting.

Developing a model that is as simple as possible but not simpler: While building the final model presented here, we started with the simplest possible representation of each process and built in complexity until the model had desired predictive capability. For example, to model the meal function in the T2DM setting, we first used an impulse function that concentrates all the ingested glucose at a single meal time instant. Numerical simulations showed that this choice was too simple to reflect reality. The source of the problem is insightful: concentrating all of the glucose in the meal at one time point, the start time of the meal, caused the corresponding simulated BG levels to increase very rapidly to very high values, which were not even on the same order as the true BG measurements, e.g., when BG measurements are in the range of $\sim 100 - 150$ mg/dl, the simulated BG values are in the range of $\sim 700 - 800$ mg/dl. Physiologically, it is likely that a sharp spike in glucose intake would cause a spike in BG, but it is also likely the spike would be narrow and BG would return to near normal values quickly; however a full discussion of the physiological effects of such a dose of glucose is beyond the scope of this discussion. We then tried a simplistic solution in which we represented nutrition ingestion as a square-wave function, which was sum of constant functions that have the value G_i/T over the interval $[t_i^{(m)}, t_i^{(m)} + T]$ where G_i is the total amount of glucose ingestion in the meal starting at time $t_i^{(m)}$, and T is a time-scale for transfer of glucose from stomach to blood. That is, we set $m(t) = \sum_{i=1}^{K_m} \frac{G_i}{T} \mathbb{1}_{[t_i^{(m)}, t_i^{(m)}+T]}(t)$, and let T be a model parameter to be estimated for each patient from data. This function produced reasonable, realistic simulation results. However, the cost function we minimize to fit the model to data (see (3.25)) exhibited discontinuities related to *discontinuous behaviour of the meal model with respect to T*. The somewhat surprising result of this discontinuity was our inability to accurately estimate glycemic responses to nutrition. Meaning, with a square wave nutrition delivery function, inference failed. These failures led us to choose a smooth function for nutrition delivery that then led to a continuous cost function with respect to the unknown model parameters. These issues led to the meal function as defined in (3.5) that satisfies both the requirements. Meaning, the model development was driven both by the need to reconcile the model with realistic physiology and by the need to be able to preform inference with data. Similar considerations applied to other aspects of model development.

Impact of the Nutrition Function Choice in the ICU Context: We also consider different form for the nutrition function in the ICU setting, in order to test robustness of our modeling to the simplistic piecewise constant model that we adopt in this case, and in view of the fact that in the T2D setting we found the need for a more sophistictaed model for nutrition uptake. First, note that because of ICU patients are tube-fed with nutrition quantities that are considerably less, per unit time, than a healthy indivudual would ingest, per unit time, over the duration of a regular meal, it is reasonable to consider the use of a piecewise constant function to model the effect of nutrition on the BG levels for ICU patients. Nonetheless we investigated if modifying the piecewise constant function as shown in Figure 3.9 could improve the parameter estimation and/or forecasting results. The idea behind this modification is to model the effect of nutrition on BG level via an initial exponential increase that reaches a maximum value and then, when the nutrition delivery stops, this effect decreases exponentially. The rate of increase and decrease are represented by two model parameters a and b. These parameters are similar to the ones that we used to model the effect of nutrition in the T2DM context, but the nutrition function is not exactly the same since the nutrition effect in these two cases have different characteristics. In addition, since the values of these two parameters can be patient-specific, using this function introduces two more parameters to be estimated in the ICU setting, increasing the flexibility and the complexity at the same time. Hence the parameters to be estimated are $G_b, \gamma, \sigma, \beta, a, b$.

The function we use has the following form:

$$m(t) = \sum_{k=1}^{K_m} c_k((1 - e^{-a(t-t_k^{(m)})})\mathbb{1}_{[t_k^{(m)}, t_{k+1}^{(m)})}(t) \\ + (1 - e^{-ah_k^{(m)}})(2 - e^{b(t-t_k^{(m+1)})})\mathbb{1}_{[t_k^{(m+1)}, t_{k+1}^{(m)} + \frac{ln(2)}{b}]}(t)), \tag{3.26}$$

where $h_k^{(m)} := t_k^{(m+1)} - t_k^{(m)}$ and

$$c_k := \frac{d_k h_k^{(m)}}{h_k^{(m)} + (e^{-ah_k^{(m)}}/a) + (1 - e^{-ah_k^{(m)}})((2ln(2) - 1)/b)},$$

is the normalizing constant for $k = 1, 2, ..., K_m$.

The numerical results concerning the forecasting capability of the MSG model with this nutrition function are summarized in Table 3.7. A quick comparison of these results with the ones in Tables 3.5 and 3.6 shows that the percentages for 1- and 2-stdev bands are a little less whereas MSE and MPE are either the same or slightly higher than the ones obtained with piecewise constant nutrition function. Hence this comparison suggests that using the more physiologically accurate version of the meal function did not, in this ICU case, introduce any improvements. Since the new function introduced two more new parameters to be estimated this also leads to a problem that is computationally more challanging, especially when the problem is prone to identification issues. Overall our work comparing the original and new meal function demonstrates that the original piecewise constant choice of the meal function is an appropriate choice in this case.

Comparison of Forecasting Efficiency with Other Models: In order to evaluate the effectiveness of the MSG model, we ran experiments comparing it with the

	1-std %	2-std %	mse	rmse	mpe	average model std
Patient 4	56.73	91.81	392.90	19.82	11.87	16.86
Patient 5	59.59	83.42	1239.70	35.21	19.68	25.25
Patient 6	57.36	84.11	2094.70	45.77	26.01	33.40

Table 3.7: Forecasting results obtained with optimization approach for the MSG model with the nutrition function given in (3.26). Even though the nutrition function could be considered as a better representation of the reality, it did not introduce improvement in the forecasting results.

LDP model (the T2DM setting) and with the ICU-MM (the ICU setting). The LDP and ICU-MM models are built to represent the co-evolution of glucose and insulin dynamics, in contrast to our simplified model which models glucose dynamics with insulin as parameterically-dependent input. For the purpose of comparison, we used $1-$ and $2-$stdev band percentages, MSE, RMSE, and MPE as the evaluation metrics. In both the T2DM ICU settings we showed that: *(i)* even though the MSG model was developed to capture the mean behavior of BG level, and the numerical scheme used for identification is not specifically designed to minimize the MSE, the MSE over the test data obtained with the MSG level is typically smaller than, and in the worst case at the same level as, the MSE obtained with the LDP model (for T2DM) and ICU-MM (for ICU); *(ii)* the confidence bands obtained via the MSG model are more effective in that they capture a higher proportion of the true BG measurements than the confidence bands found from the LDP model and the ICU-MM. This is likely due in part to the fact that the MSG model uses a stochastic description to encapsulate possible fluctuations around the mean in the quanity of interest, which is the BG measurements; the LDP and ICU-MM are deterministic models and fluctuations are captured through the ensemble method used to fit the data. These two points show that the MSG model is at least as effective in forecasting future BG levels as the LDP and ICU-MM models, in the T2DM and ICU settings; indeed it is typically more effective. Achieving this level of accuracy in the two different settings is achieved by using a simple model, apprpropriate for the available data, but complex enough to be interpretable and to capture the underlying physiology. The resulting simple MSG model has a smaller number of unknown parameters than do the LDP and ICU-MM models, therby providing more robust estimation and inference results.

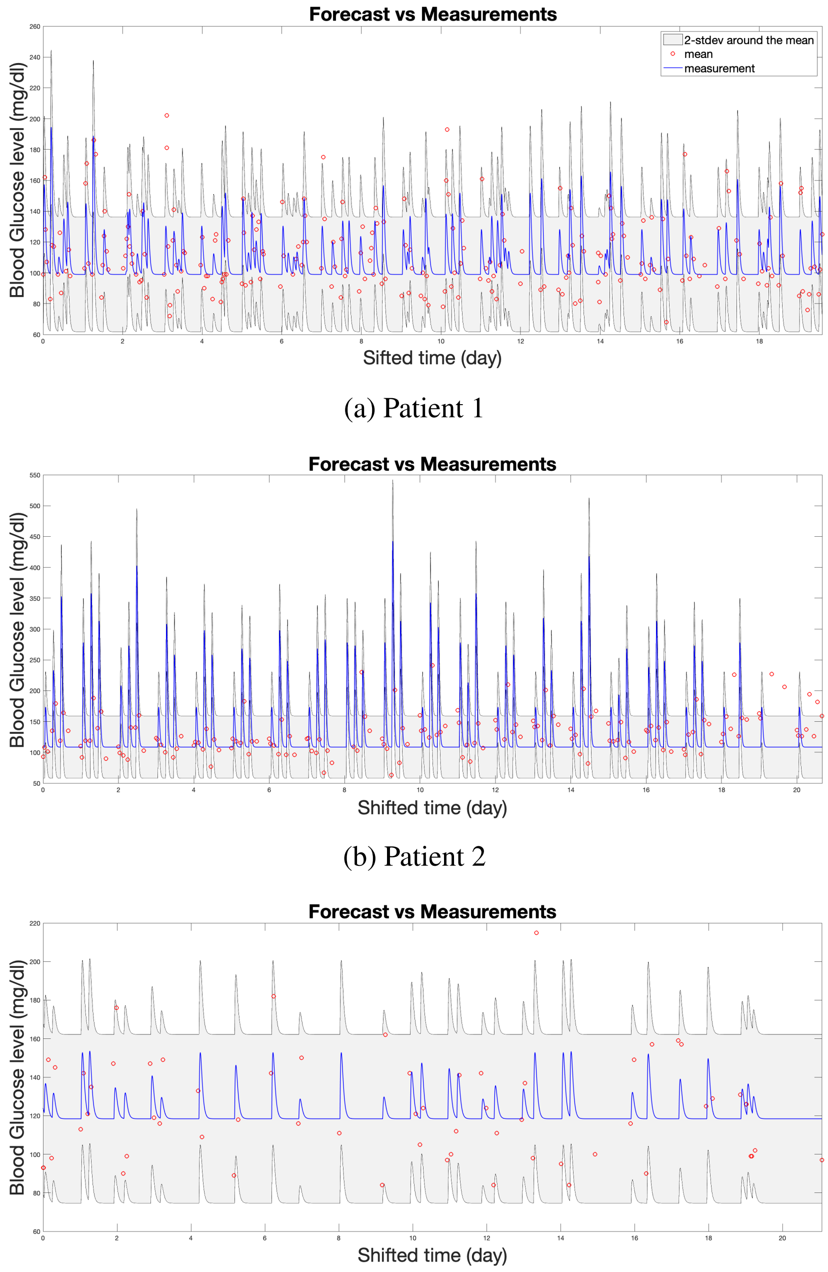
Generalizability of parameter estimation: Finally, a careful investigation of the estimated parameters and simulated BG levels in the ICU context shows that we can estimate parameter values that represent the BG levels very well when the true

BG measurements are interpretable with the model (3.12). That is, if measured BG values are responsive to the changes in the rate of nutrition and insulin IV, then the BG simulations with the estimated parameters based on this data provide a very good representation of the dynamics. However, if the BG level behavior is not driven by the nutrition and the insulin IV rates, i.e., if its response is driven by other factors such as stress-induced counter-regulatory hormone levels, then the model-estimated mean is estimated to be almost-constant. This mean estimate is still good as a representation of the average of BG measurements and the variance of the measurements are still estimated accurately enough that the 2-stdev band around the mean envelopes nearly every BG measurement. For all patients in all disease cases, independent of parameter estimation complexities, we obtain good estimates of the forecast mean and variance of the BG levels we achieve with the model are likely accurate enough to be helpful in clinical settings.

Outlook: The model we have developed has demonstrable predictive capability and discriminates between datasets in a patient-specific manner. Yes it has some limitations, which give space for future development, and also suggests some natural next-step applications. We outline a number of possible future directions. *Glycemic control:* Given the MSG model construction, an obvious next step is to formulate the work on the control problem where we determine estimates of the input ranges of nutrition and insulin, necessary to keep the output, here BG, in a desired target range. This is a similar approach to something like the artificial pancreas/beta-cell project, but the inputs would include nutrition, the settings would include T2DM and ICU glycemic management, and the goal would not be a closed loop but rather an open loop system. *Parameter estimation short-comings and advancement:* In T2DM setting, the estimation results with optimization and MCMC approaches for the parameters a and b used to define the rate of appearance and absorption of glucose produce conflicting results. In the ICU setting, we observe some identifiability issues for the parameters modeling glucose removal with body's own effort and with insulin IV. We plan to address these issues in future. Key questions are whether different parameter estimation techniques can resolve the problems, or whether further data is required, and if so which data, and more fundamentally whether the model used is appropriate for the data. A related issue is the possibility of using mixed effects models [395, 418] in order to share common information in different patient data sets, whilst also retaining the advantages of patient specific learning. *Comparison with more complex models:* In order to have a better understanding about the effectiveness of this model to encapsulate BG dynamics and resolve the mean and variance of BG levels, we plan to compare it with more complex models, such as a second order linear SDE (which would allow for oscillatory dynamics but retain the advantages of linearity and Gaussianity exploited here) and the Ultradian model [399] (which is a widely- accepted physiogically based model). Such a comparison would happen within design similar to what we used in this paper for both T2DM and ICU context. Furthermore in the situation where control machinery has been added to the model, we can evaluate the various model's effectiveness in a control-based setting. *Phenotyping:* Because the parameters of the MSG model are interpretable and track physiology reasonable well, we could potentially use the model parameter estimates for phenotyping studies, [6, 7, 8]. Meaning, we could estimate parameter for individuals in a given health state, establishing an inferred phenotype for the patient, and then relate this phenotype to other external health features or cluster the patient phenotypes in an effort to find structure among the inferred physiology. We have deemed efforts such as this high-fidelity phenotyping [184] and believe that this model has the potential to be used to these ends. *Exploiting model error to understand physiology:* It is known that BG levels are mainly driven by the carbohydrates, however, there are also other factors that impact glucose levels. A partial list of particularly interesting features that impact BG levels and are of practical interest include macro-nutrients other than carbohydrates, exercise, sleep, and stress levels of patients. The presence of these features will induce systematic forecasting errors allowing us to use machine learning to explore the statistical relationship between these factors and BG levels. This would give us a systematic platform for potentially furthering the understanding of the glucose-insulin system and result in more accurate parameter estimation and forecasting. *Further model generalization to include other glucose-data driven situations:* We have not investigated how the MSG model might work given oral glucose tolerance test (OGTT) data. The OGTT is one of the standard settings for glucose-insulin model development and potential use; we know of only one model that currently generalized to both OGTT and clinical data [158] and we would like to add the MSG model to this list. *T1DM:* We have a initial version of the MSG model that could be used within T1DM setting. It would be interest to test this version on T1DM data. Since the time-scales of health progression here are more similar to those of T2DM than the ICU setting, giving hope that the method might have similar predictive capability in this setting. Because of not having access to such a T1DM dataset, we haven't been able to work with this version in this paper. We plan to pursue a number of the research directions outlined here in the immediate future.

Acknowledgements

We acknowledge financial support from NIH RO1 LM012734 “Mechanistic Machine Learning”. DA acknowledges helpful discussions with Bruce Gluckman, Rammah Abohtyra and Cecilia Diniz Behn.

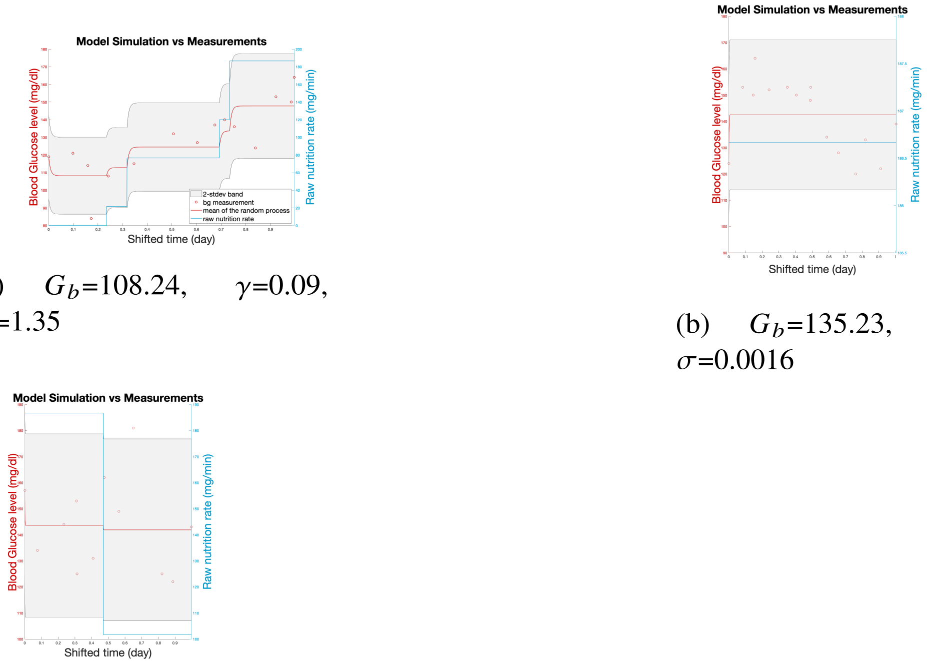


(a) Patient 1

(b) Patient 2

(c) Patient 3

Figure 3.3: Forecasting results in the T2DM setting obtained via models formed by using the estimated parameters with the optimization approach. In each plot, the red circles show the true BG measurements, the blue curve shows the mean of the model output, and the gray region is the estimated 2-stdev band around the mean of the model output. These forecasting results show that the proposed model mean, when equipped with confidence bands found from standard deviations, estimate the BG levels accurately, and in a patient specific way. This reinforces the claim that the model parameters could be used to provide information about the health condition of individual patients.

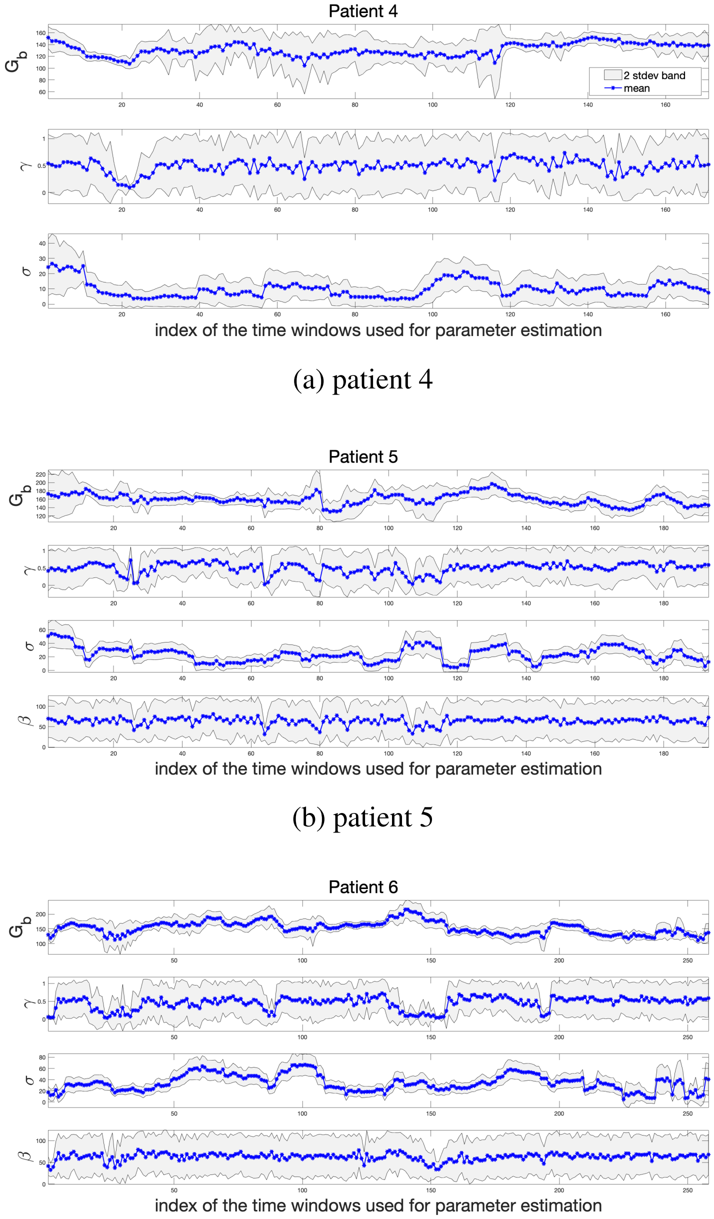


(a) G_b=108.24, γ=0.09, σ=1.35

(b) G_b=135.23, γ=0.51, σ=0.0016

(c) G_b=139.87, γ=1.00, σ=10.17

Figure 3.4: BG simulations are shown with respective to the estimated parameters over training time window. In each plot, the light blue curve is the glucose rate in the nutrition delivered to the patient (right y-axis), the red circles show the true BG measurements (left y-axis), the red curve is the mean of the model output (left y-axis), and the gray area is the 2-stdev band around the mean of the model output (left y-axis). These figures show two main cases that could arise as a result of parameter estimation in the ICU setting: Figure 3.4a: The input (nutrition rate) is reflected in the output (BG measurements), Figures 3.4b and 3.4c: The input is not reflected in the output, which makes it impossible to estimate the decay rate γ.

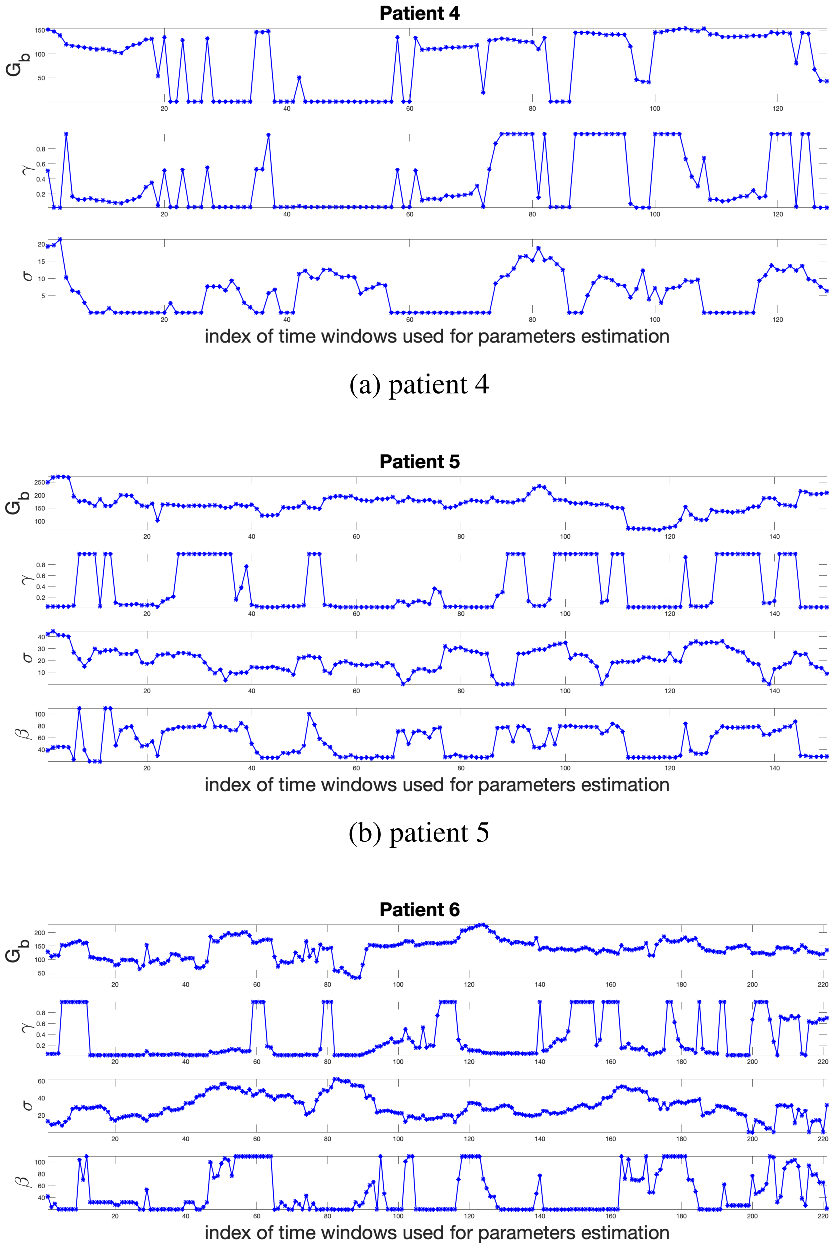


(a) patient 4

(b) patient 5

(c) patient 6

Figure 3.5: Parameter estimation and uncertainty quantification results in the ICU setting obtained with MCMC approach. In each plot, the blue stars represent the point-estimate of each parameter (mean of the resulting random samples) and the gray area is the 2-stdev band around the point-estimates (also obtained from the resulting random samples). These results show that the estimated model parameters exhibit biophysically realistic values and change relatively smoothly; this is to be expected since the consecutive (moving) time windows (each of length around one day) have a large overlap. However, the cases where there is a considerable change in the estimated parameter are also understandable because of rapid changes in the patients' health condition and/or the difficulty in extracting such information due to patients' glycemic response. On the other hand, 1- and 2-stdev bands enforces the reliability of the estimated parameters, especially, G_b and σ, which are the most important parameters in predicting the mean and variance of BG levels.

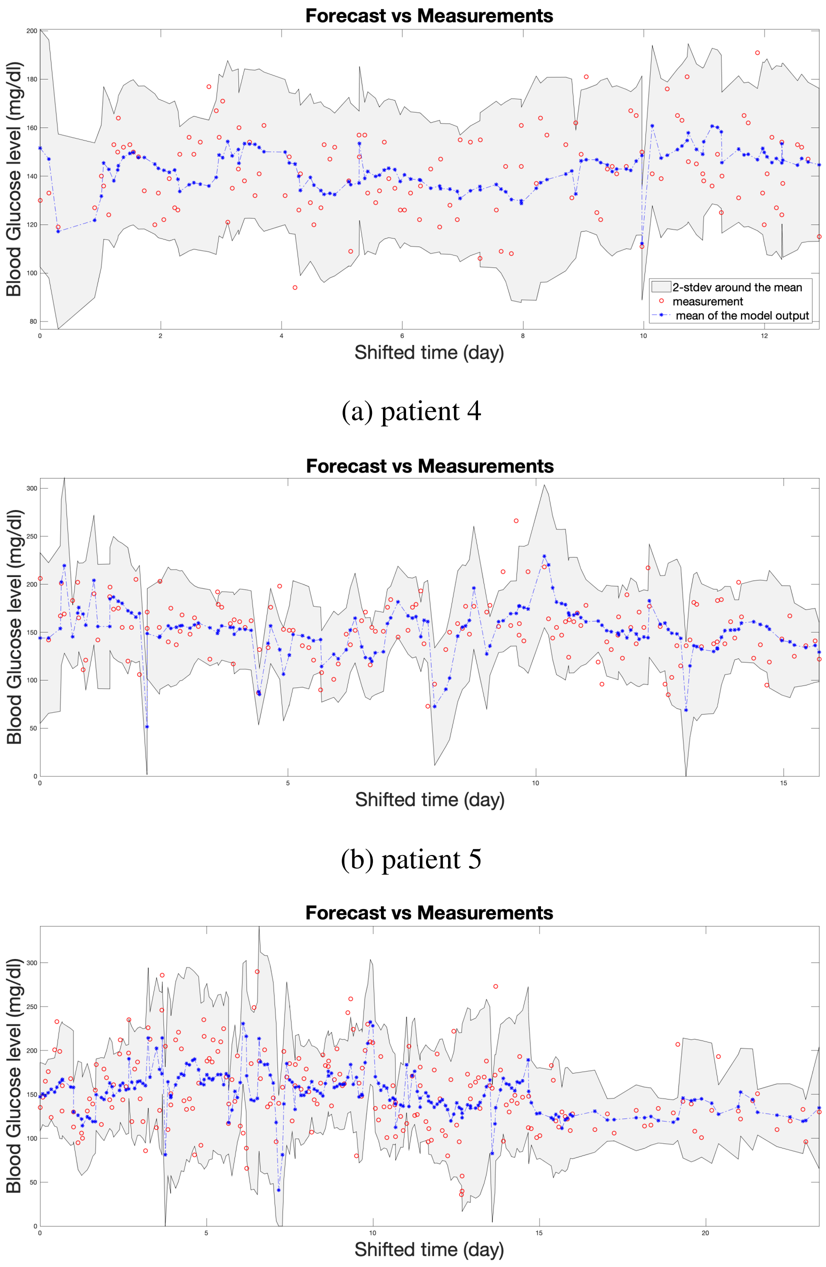


(a) patient 4

(b) patient 5

(c) patient 6

Figure 3.6: Parameter estimation results in the ICU setting obtained with the optimization approach. In each plot, the blue stars represent the MAP estimator of the corresponding model parameter. These results provide important understanding of the system through the cases whether the data is interpretable through the model or not. When both of the basal glucose value, G_b and the decay rate, γ attain physiologically plausible values, this is mostly representative of a case where the data is interpretable through the model, whereas other cases reflect when it is not possible to estimate the decay rate and how this situation propagates through the other estimated parameters.

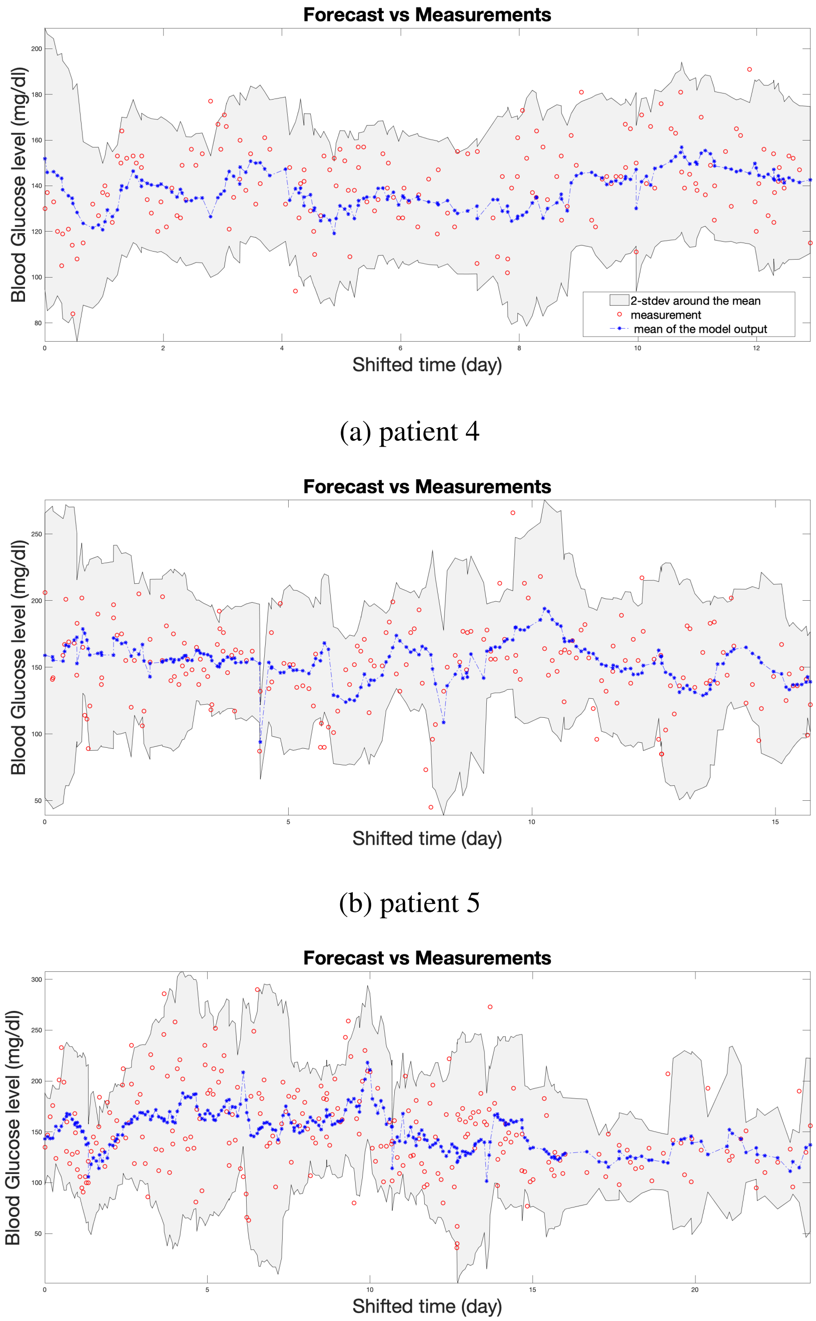


(a) patient 4

(b) patient 5

(c) patient 6

Figure 3.7: Forecasting results obtained based on parameters estimated with optimization, in the ICU setting. In each plot, the red circles show the true BG measurements, the blue stars show the mean of the model output, and the gray region shows the 2-stdev band around this mean. For all three patients, the forecasted mean captures the actual behavior of BG levels (not used in the training of parameters). Moreover, the 2-stdev band narrows down over the time periods on which BG levels are relatively stable, and widen over the intervals where the BG value has larger variance. Capturing this behavior with reasonably tightly confidence bands is a very useful and valuable feature of the proposed model.



(a) patient 4

(b) patient 5

(c) patient 6

Figure 3.8: Forecasting results obtained based on parameters estimated with MCMC, in the ICU setting. In each plot, the red circles show the true BG measurements, the blue stars show the mean of the model output, and the gray region shows the 2-stdev band around this mean. These results are, in general, very close to those obtained using the optimization approach, and the most relevant properties are shared by them both. Obtaining similar results with another numerical solution technique based on the same mechanistic model shows the reliability of the model and estimated model parameters.

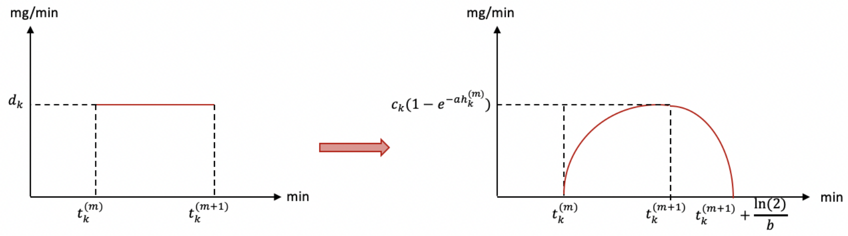


Figure 3.9: Smoothing piecewise constant nutrition function that is used for ICU patients

Chapter 4

A FRAMEWORK FOR MACHINE LEARNING OF MODEL ERROR IN DYNAMICAL SYSTEMS

Remark 4.0.1. This chapter is derived from the manuscript by Levine and Stuart [238] published in Communications of the American Mathematical Society.

4.1 Introduction

4.1.1 Background and Literature Review

The modeling and prediction of dynamical systems and time-series is an important goal in numerous domains, including biomedicine, climatology, robotics, and the social sciences. Traditional approaches to modeling these systems appeal to careful study of their mechanisms, and the design of targeted equations to represent them. These carefully built *mechanistic models* have impacted humankind in numerous arenas, including our ability to land spacecraft on celestial bodies, provide high-fidelity numerical weather prediction, and artificially regulate physiologic processes, through the use of pacemakers and artificial pancreases, for example. This paper focuses on the learning of *model error*: we assume that an imperfect mechanistic model is known, and that data are used to improve it. We introduce a framework for this problem, focusing on distinctions between Markovian and non-Markovian model error, providing a unifying review of relevant literature, developing some underpinning theory related to both the Markovian and non-Markovian settings, and presenting numerical experiments which illustrate our key findings.

To set our work in context, we first review the use of data-driven methods for time-dependent problems, organizing the literature review around four themes comprising Sections 4.1.1.1 to 4.1.1.3 and 4.1.1.5; these are devoted, respectively, to pure data-driven methods, hybrid methods that build on mechanistic models, non-Markovian models that describe memory, and applications of the various approaches. Having set the work in context, in Section 4.1.2 we detail the contributions we make, and describe the organization of the paper.

4.1.1.1 Data-Driven Modeling of Dynamical Systems

A recent wave of machine learning successes in data-driven modeling, especially in imaging sciences, has shown that we can demand even more from existing models, or that we can design models of more complex phenomena than heretofore. Traditional models built from, for example, low order polynomials and/or linearized model reductions, may appear limited when compared to the flexible function approximation frameworks provided by neural networks and kernel methods. Neural networks, for example, have a long history of success in modeling dynamical systems [294, 148, 217, 350, 348, 351, 223] and recent developments in deep learning for operators continue to propel this trend [268, 40, 245, 244].

The success of neural networks arguably relies on balanced expressivity and generalizability, but other methods also excel in learning parsimonious and generalizable representations of dynamics. A particularly popular methodology is to perform sparse regression over a dictionary of vector fields, including the use of thresholding approaches (SINDy) [61] and L_1-regularized polynomial regression [408, 368, 367, 369]. Non-parametric methods, like Gaussian process models [342], have also been used widely for modeling nonlinear dynamics [423, 138, 213, 86]. While a good choice of kernel is often essential for the success of these methods, recent progress has been made towards automatic hyperparameter tuning via parametric kernel flows [162]. Successes with Gaussian process models were also extended to high dimensional problems by using random feature map approximations [335] within the context of data-driven learning of parametric partial differential equations (PDEs) and solution operators [295]. Advancements to data-driven methods based on Koopman operator theory and dynamic mode decomposition also offer exciting new possibilities for predicting nonlinear dynamics from data [410, 214, 14].

It is important to consider whether to model in discrete- or continuous-time, as both have potential advantages. The primary positive for continuous-time modeling lies in its flexibility and interpretability. In particular, continuous-time approaches are more readily and naturally applied to irregularly sampled timeseries data, e.g. electronic health record data [358], than discrete-time methods. Furthermore, this flexibility with respect to timestep enables simple transferability of a model learnt from discrete-time data at one timestep, to new settings with a different timestep and indeed to variable timestep settings; the learned right-hand-side can be used to generate numerical solutions at any timestep. On the other hand, applying a discrete-time model to a new timestep either requires exact alignment of subsampled data or some post-processing interpolation step. Continuous-time models may also provide greater interpretability than discrete-time methods when the right-hand-side of the ordinary differential equation (ODE) is a more physically interpretable object than the Δt-solution operator (e.g. for equation discovery, [196]).

Traditional implementations of continuous-time learning require accurate estimation of time-derivatives of the state, but this may be circumvented using approaches that leverage autodifferentiation software [304, 358, 195] or methods which learn from statistics derived from time-series, such as moments or correlation functions [375]. Keller and Du [204] and Du et al. [114] provide rigorous analysis demonstrating how inference of a continuous-time model from discrete-time data must be conducted with great care; they prove how stable and consistent linear multistep methods for continuous-time integration may not possess the same guarantees when used for the inverse problem, i.e. discovery of dynamics. Queiruga et al. [332] provide pathological illustrations of this phenomenon in the context of Runge-Kutta methods.

Discrete-time approaches, on the other hand, are easily deployed when train and test data sample rates are the same. For applications in which data collection is easily configured (e.g. simulated settings, available automatic sensors, etc.), discrete-time methods are typically much easier to implement and test than continuous-time methods. Moreover, they allow for “non-intrusive” model correction, as additions are applied outside of the numerical integrator; this may be relevant for practical integration with complex simulation software. In addition, discrete-time approaches can be preferable when there is unavoidably large error in continuous-time inference [90, 266].

Both non-parametric and parametric model classes are used in the learning of dynamical systems, with the latter connecting to the former via the representer theorem, when Gaussian process regression [342] is used [66, 145, 165].

4.1.1.2 Hybrid Mechanistic and Data-Driven Modeling

Attempts to transform domains that have relied on traditional mechanistic models, by using purely data-driven (i.e. *de novo* or “learn from scratch”) approaches, often fall short. Now, there is a growing recognition by machine learning researchers that these mechanistic models are very valuable [289], as they represent the distillation of centuries of data collected in countless studies and interpreted by domain experts. Recent studies have consistently found advantages of hybrid methods that blend mechanistic knowledge and data-driven techniques; Willard et al. [434] provide a thorough review of this shift amongst scientists and engineers. Not only do these hybrid methods improve predictive performance [316], but they also reduce data demands [333] and improve interpretability and trustworthiness, which is essential for many applications. This is exemplified by work in autonomous drone landing [382] and helicopter flying [334], as well as predictions for COVID-19 mortality risk [392] and COVID-19 treatment response [331].

The question of how best to use the power of data and machine learning to leverage and build upon our existing mechanistic knowledge is thus of widespread current interest. This question and research direction has been anticipated over the last thirty years of research activity at the interface of dynamical systems with machine learning [350, 437, 263], and now a critical mass of effort is developing. A variety of studies have been tackling these questions in weather and climate modeling [203, 131]; even in the imaging sciences, where pure machine learning has been spectacularly successful, emerging work shows that incorporating knowledge of underlying physical mechanisms improves performance in a variety of image recognition tasks [25].

As noted and studied by Ba, Zhao, and Kadambi [25] and Freno and Carlberg [137] and others, there are a few common high-level approaches for blending machine learning with mechanistic knowledge: (1) use machine learning to learn additive residual corrections for the mechanistic model [364, 382, 196, 165, 131, 267, 264, 446]; (2) use the mechanistic model as an input or feature for a machine learning model [316, 233, 46]; (3) use mechanistic knowledge in the form of a differential equation as a final layer in a neural network representation of the solution, or equivalently define the loss function to include approximate satisfaction of the differential equation [339, 340, 81, 391]; and (4) use mechanistic intuition to constrain or inform the machine learning architecture [160, 282]. Many other successful studies have developed specific designs that further hybridize these and other perspectives [161, 137, 445, 192]. In addition, parameter estimation for mechanistic models is a well-studied topic in data assimilation, inverse problems, and other mechanistic modeling communities, but recent approaches that leverage machine learning for this task may create new opportunities for accounting for temporal parameter variations [287] and unknown observation functions [255].

An important distinction should be made between physics-informed *surrogate* modeling and what we refer to as *hybrid* modeling. Surrogate modeling primarily focuses on replacing high-cost, high-fidelity mechanistic model simulations with similarly accurate models that are cheap to evaluate. These efforts have shown great promise by training machine learning models on expensive high-fidelity simulation data, and have been especially successful when the underlying physical (or other domain-specific) mechanistic knowledge and equations are incorporated into the model training [339] and architecture [282]. We use the term hybrid modeling, on the other hand, to indicate when the final learned system involves interaction (and possibly feedback) between mechanism-based and data-driven models [316].

In this work, we focus primarily on hybrid methods that learn residuals to an imperfect mechanistic model. We closely follow the discrete-time hybrid modeling framework developed by [165], while providing new insights from the continuous-time modeling perspective. The benefits of this form of hybrid modeling, which we and many others have observed, are not yet fully understood in a theoretical sense. Intuitively, nominal mechanistic models are most useful when they encode key nonlinearities that are not readily inferred using general model classes and modest amounts of data. Indeed, classical approximation theorems for fitting polynomials, fourier modes, and other common function bases directly reflect this relationship by bounding the error with respect to a measure of complexity of the target function (e.g. Lipschitz constants, moduli of continuity, Sobolev norms, etc.) [107][Chapter 7]. Recent work by E, Ma, and Wu [120] provides a priori error bounds for two-layer neural networks and kernel-based regressions, with constants that depend explicitly on the norm of the target function in the model-hypothesis space (a Barron space and a reproducing kernel Hilbert space, resp.). At the same time, problems for which mechanistic models only capture low-complexity trends (e.g. linear) may still be good candidates for hybrid learning (over purely data-driven), as an accurate linear model reduces the parametric burden for the machine-learning task; this effect is likely accentuated in data-starved regimes. Furthermore, even in cases where data-driven models perform satisfactorily, a hybrid approach may improve interpretability, trustworthiness, and controllability without sacrificing performance.

Hybrid models are often cast in Markovian, memory-free settings where the learned dynamical system (or its learned residuals) are solely dependent on the observed states. This approach can be highly effective when measurements of all relevant states are available or when the influence of the unobserved states is adequately described by a function of the observables. This is the perspective employed by Shi et al. [382], where they learn corrections to physical equations of motion for an autonomous vehicle in regions of state space where the physics perform poorly—these residual errors are *driven* by un-modeled turbulence during landing, but can be *predicted* using the observable states of the vehicle (i.e. position, velocity, and acceleration). This is also the perspective taken in applications of high-dimensional multiscale dynamical systems, wherein sub-grid closure models parameterize the effects of expensive fine-scale interactions (e.g. cloud simulations) as functions of the coarse variables [153, 207, 403, 53, 301, 344, 375, 37]. The result is a hybrid dynamical system with a physics-based equation defined on the coarse variables with a Markovian correction term that accounts for the effects of the expensive fine scale dynamics.

4.1.1.3 Non-Markovian Data-Driven Modeling

Unobserved and unmodeled processes are often responsible for model errors that cannot be represented in a Markovian fashion within the observed variables alone. This need has driven substantial advances in memory-based modeling. One approach to this is the use of delay embeddings [402]. These methods are inherently tied to discrete time representations of the data and, although very successful in many applied contexts, are of less value when the goal of data-driven learning is to fit continuous-time models; this is a desirable modeling goal in many settings.

An alternative to understanding memory is via the Mori-Zwanzig formalism, which is a fundamental building block in the presentation of memory and hidden variables and may be employed for both discrete-time and continuous-time models. Although initially developed primarily in the context of statistical mechanics, it provides the basis for understanding hidden variables in dynamical systems, and thus underpins many generic computational tools applied in this setting [89, 462, 152]. It has been successfully applied to problems in fluid turbulence [117, 311] and molecular dynamics [242, 176]. Lin and Lu [253] demonstrate connections between Mori-Zwanzig and delay embedding theory in the context of non-linear autoregressive models using Koopman operator theory. Indeed, Gilani, Giannakis, and Harlim [145] show a correspondence between the Mori-Zwanzig representation of the Koopman operator and Taken's delay-embedding flow map. Studies by Ma, Wang, and E [274] and Wang, Ripamonti, and Hesthaven [427] demonstrate how the Mori-Zwanzig formalism motivates the use of recurrent neural networks (RNNs) [361, 149] as a deep learning approach to non-Markovian closure modeling. Harlim et al. [165] also use the Mori-Zwanzig formalism to deduce a non-Markovian closure model, and evaluate RNN-based approximations of the closure dynamics. Closure modeling using RNNs has recently emerged as a new way to learn memory-based closures

[202, 75, 165].

Although the original formulation of Mori-Zwanzig as a general purpose approach to modeling partially observed systems was in continuous-time [89], many practical implementations adopt a discrete-time picture [101, 90, 253]. This causes the learned memory terms to depend on sampling rates, which, in turn, can inhibit flexibility and interpretability.

Recent advances in continuous-time memory-based modeling, however, may be applicable to these non-Markovian hybrid model settings. The theory of continuous-time RNNs (i.e. formulated as differential equations, rather than a recurrence relation) was studied in the 1990s [139, 30], albeit for equations with a specific additive structure. This structure was exploited in a continuous-time reservoir computing (RC) approach by Lu, Hunt, and Ott [270] for reconstructing chaotic attractors from data. Comparisons between RNNs and RC (a subclass of RNNs with random parameters fixed in the recurrent state) in discrete-time have yielded mixed conclusions in terms of their relative efficiencies and ability to retain memory [330, 142, 421, 74]. Recent formulations of continuous-time RNNs have departed slightly from the additive structure, and have focused on constraints and architectures that ensure stability and accuracy of the resulting dynamical system [72, 125, 299, 358, 381, 304]. In addition, significant theoretical work has been performed for linear RNNs in continuous-time [243]. Nevertheless, these various methods have not yet been formulated within a hybrid modeling framework, nor has their approximation power been carefully evaluated in that context. A recent step in this direction, however, is the work by Gupta and Lermusiaux [157], which tackles non-Markovian hybrid modeling in continuous-time with neural network-based delay differential equations (DDEs).

4.1.1.4 Noisy Observations and Data Assimilation

For this work we consider settings in which the observations may be both noisy and partial; the observations may be partial either because the system is undersampled in time, or because certain variables are not observed at all. We emphasize that ideas from statistics can be used to smooth and/or interpolate data to remove noise and deal with undersampling [94] and to deal with missing data [285]; and ideas from data assimilation [21, 227, 346] can be used to remove noise and to learn about unobserved variables [82, 151, 150]. In some of our experiments we will use noise-free data in continuous-time, to clearly expose issues separate from noise/interpolation; but in other experiments we will use methodologies from data assimilation to enhance our learning [82].

4.1.1.5 Applications of Data-Driven Modeling

In order to deploy hybrid methods in real-world scenarios, we must also be able to cope with noisy, partial observations. Accommodating the learning of model error in this setting, as well as state estimation, is an active area of research in the data assimilation (DA) community [329, 131, 41]. Learning dynamics from noisy data is generally non-trivial for nonlinear systems—there is a chicken-and-egg problem in which accurate state estimation typically relies on the availability of correct models, and correct models are most readily identified using accurate state estimates. Recent studies have addressed this challenge by attempting to jointly learn the noise and the dynamics. Gottwald and Reich [151] approach this problem from a data assimilation perspective, and employ an Ensemble Kalman Filter (EnKF) to iteratively update the parameters for their dynamics model, then filter the current state using the updated dynamics. A recent follow-up to this work applies the DA-approach to partially-observed systems, and learns a model on a space of discrete-time delay-embeddings [150]. Similar studies were performed by Brajard et al. [50], and applied specifically in model error scenarios [49, 131, 432]. Ayed et al. [24] focus on learning a continuous-time neural network representation of an ODE from partial observations, and learn a separate encoder neural network to map a historical warmup sequence to likely initial conditions in the un-observed space. Kaheman et al. [196] approach this problem from a variational perspective, performing a single optimization over all noise sequences and dynamical parameterizations. Nguyen et al. [298] use an Expectation-Maximization (EM) perspective to compare these variational and ensemble-based approaches, and further study is needed to understand the trade-offs between these styles of optimization. Chen, Sanz-Alonso, and Willett [82] study an EnKF-based optimization scheme that performs joint, rather than EM-based learning, by running gradient descent on an architecture that backpropagates through the data assimilator.

We note that data assimilators are themselves dynamical systems, which can be tuned (using optimization and machine learning) to provide more accurate state updates and more efficient state identification. However, while learning improved DA schemes is sometimes viewed as a strategy for coping with model error [461], we see the optimization of DA and the correction of model errors as two separate problems that should be addressed individually.

When connecting models of dynamical systems to real-world data, it is also essential to recognize that available observables may live in a very different space than the underlying dynamics. Recent studies have shown ways to navigate this using autoencoders and dynamical systems models to jointly learn a latent embedding and dynamics in that latent space [71]. Proof of concepts for similar approaches primarily focus on image-based inputs, but have potential for applications in medicine [255] and reduction of nonlinear PDEs [282].

4.1.2 Our Contributions

Despite this large and recent body of work in data-driven learning methods and hybrid modeling strategies, many challenges remain for understanding how to best combine mechanistic and machine-learned models; indeed, the answer is highly dependent on the application. Here, we construct a mathematical framework that unifies many of the common approaches for blending mechanistic and machine learning models; having done so we provide strong evidence for the value of hybrid approaches. Our contributions are listed as follows:

1. We provide an overarching framework for learning model error from (possibly noisy) data in dynamical systems settings, studying both discrete- and continuous-time models, together with both memoryless (Markovian) and memory-dependent representations of the model error. This formulation is agnostic to choice of mechanistic model and class of machine learning functions.

2. We study the Markovian learning problem in the context of ergodic continuous-time dynamics, proving bounds on excess risk and generalization error.

3. We present a simple approximation theoretic approach to learning memory-dependent (non-Markovian) model error in continuous-time, proving a form of universal approximation for two families of memory-dependent model error defined using recurrent neural networks.

4. We describe numerical experiments which: a) demonstrate the utility of learning model error in comparison both with pure data-driven learning and with pure (but slightly imperfect) mechanistic modeling; b) compare the benefits of learning discrete- versus continuous-time models; c) demonstrate the utility of auto-differentiable data assimilation to learn dynamics from partially observed, noisy data; d) explain issues arising in memory-dependent

model error learning in the (typical) situation where the dimension of the memory variable is unknown.

In Section 4.2, we address contribution 1. by defining the general settings of interest for dynamical systems in both continuous- and discrete-time. We then link these underlying systems to a machine learning framework in Sections 4.3 and 4.4; in the former we formulate the problem in the setting of statistical learning, and in the latter we define concrete optimization problems found from finite parameterizations of the hypothesis class in which the model error is sought. Section 4.5 is focused on specific choices of architectures, and underpinning theory for machine learning methods with these choices: we analyze linear methods from the perspective of learning theory in the context of ergodic dynamical systems (contribution 2.); and we describe an approximation theorem for continuous-time hybrid recurrent neural networks (contribution 3.). Finally, Section 4.6 presents our detailed numerical experiments; we apply the methods in Section 4.5 to exemplar dynamical systems of the forms outlined in Section 4.2, and highlight our findings (contribution 4.).

4.2 Dynamical Systems Setting

In the following, we use the phrase *Markovian model error* to describe model error expressible entirely in terms of the observed variable at the current time, the memoryless situation; *non-Markovian model error* refers to the need to express the model error in terms of the past history of the observed variable.

We present a general framework for modeling a dynamical system with Markovian model error, first in continuous-time (Section 4.2.1) and then in discrete-time (Section 4.2.2). We then extend the framework to the setting of non-Markovian model error (Section 4.2.3), including a parameter ε which enables us to smoothly transition from scale-separated problems (where Markovian closure is likely to be accurate) to problems where the unobserved variables are not scale-separated from those observed (where Markovian closure is likely to fail and memory needs to be accounted for).

It is important to note that the continuous-time formulation necessarily assumes an underlying data-generating process that is continuous in nature. The discrete-time formulation can be viewed as a discretization of an underlying continuous system, but can also represent systems that are truly discrete.

The settings that we present are all intended to represent and classify common situations that arise in modeling and predicting dynamical systems. In particular, we stress two key features. First, we point out that mechanistic models (later referred to as a vector field f_0 or flow map Ψ_0) are often available and may provide predictions with reasonable fidelity. However, these models are often simplifications of the true system, and thus can be improved with data-driven approaches. Nevertheless, they provide a useful starting point that can reduce the complexity and data-hunger of the learning problems. In this context, we study trade-offs between discrete- and continuous-time framings. While we begin with fully-observed contexts in which the dynamics are Markovian with respect to the observed state x, we later note that we may only have access to partial observations x of a larger system (x, y). By restricting our interest to prediction of these observables, we show how a latent dynamical process (e.g. a RNN) has the power to reconstruct the correct dynamics for our observables.

4.2.1 Continuous-Time

Consider the following dynamical system

$$\dot{x} = f^\dagger(x), \quad x(0) = x_0, \tag{4.1}$$

and define $\mathsf{X}_s := C([0, s]; \mathbb{R}^{d_x})$. If $f^\dagger \in C^1(\mathbb{R}^{d_x}; \mathbb{R}^{d_x})$ then (4.1) has solution $x(\cdot) \in \mathsf{X}_T$ for any $T < T_{\max} = T_{\max}(x_0) \in \mathbb{R}^+$, the maximal interval of existence.

The primary model error scenario we envisage in this section is one in which the vector field $f^\dagger$ can only be partially known or accessed: we assume that

$$f^\dagger = f_0 + m^\dagger$$

where f_0 is known to us and $m^\dagger$ is not known. For any $f_0 \in C^1(\mathbb{R}^{d_x}; \mathbb{R}^{d_x})$ (regardless of its fidelity), there exists a function $m^\dagger(x) \in C^1(\mathbb{R}^{d_x}; \mathbb{R}^{d_x})$ such that (4.1) can be rewritten as

$$\dot{x} = f_0(x) + m^\dagger(x). \tag{4.2}$$

However, for this paper, it is useful to think of $m^\dagger$ as being small relative to f_0; the function $m^\dagger$ accounts for *model error.* While the approach in (4.2) is targeted at learning residuals of f_0, $f^\dagger$ can alternatively be reconstructed from f_0 through a different function $m^\dagger(x) \in C^1(\mathbb{R}^{2d_x}; \mathbb{R}^{d_x})$ using the form

$$\dot{x} = m^\dagger(x, f_0(x)). \tag{4.3}$$

Both approaches are defined on spaces that allow perfect reconstruction of $f^\dagger$. However, the first formulation hypothesizes that the missing information is additive; the second formulation provides no such indication. Because the first approach ensures substantial usage of f_0, it has advantages in settings where f_0 is trusted by practitioners and model explainability is important. The second approach will likely see advantages in settings where there is a simple non-additive form of model error, including coordinate transformations and other (possibly state-dependent) nonlinear warping functions of the nominal physics f_0. Note that the use of f_0 in representing the model error in the augmented-input setting of (4.3) includes the case of not leveraging f_0 at all. It is, hence, potentially more useful than simply adopting an $x-$dependent model error; but it requires learning a more complex function.

The augmented-input method also has connections to model stacking [439] or bagging [51]; this perspective can be useful when there are N model hypotheses:

$$\dot{x} = m^\dagger\big(x, f_0^{(1)}(x), \ \dots\ f_0^{(N)}(x); \theta\big).$$

The residual-based design in (4.2) relates more to model boosting [371].

Our goal is to use machine learning to approximate these corrector functions $m^\dagger$ using our nominal knowledge f_0 and observations of a trajectory $\{x(t)\}_{t=0}^T \in \mathsf{X}_T$, for some $T < T_{\max}(x_0)$, from the true system (4.1). In this work, we consider only the case of learning $m^\dagger(x)$ in equation (4.2). For now the reader may consider $\{x(t)\}_{t=0}^T$ given without noise so that, in principle, $\{\dot{x}(t)\}_{t=0}^T$ is known and may be leveraged. In practice this will not be the case, for example if the data are high-frequency but discrete in time; we address this issue in what follows.

4.2.2 Discrete-Time

Consider the following dynamical system

$$x_{k+1} = \Psi^\dagger(x_k) \tag{4.4}$$

and define $\mathsf{X}_K := \ell^\infty\big(\{0, \dots, K\}; \mathbb{R}^{d_x}\big)$. If $\Psi^\dagger \in C(\mathbb{R}^{d_x}; \mathbb{R}^{d_x})$, the map yields solution $\{x_k\}_{k\in\mathbb{Z}^+} \in \mathsf{X}_\infty := \ell^\infty\big(\mathbb{Z}^+; \mathbb{R}^{d_x}\big)$.[1] As in the continuous-time setting, we assume we only have access to an approximate mechanistic model $\Psi_0 \in C(\mathbb{R}^{d_x}; \mathbb{R}^{d_x})$, which can be corrected using an additive residual term $m^\dagger \in C(\mathbb{R}^{d_x}; \mathbb{R}^{d_x})$:

$$x_{k+1} = \Psi_0(x_k) + m^\dagger(x_k), \tag{4.5}$$

[1] Here we define $\mathbb{Z}^+ = \{0, \dots, \}$, the non-negative integers, including zero.

or by feeding Ψ_0 as an input to a corrective warping function $m^\dagger \in C(\mathbb{R}^{2d_x};\mathbb{R}^{d_x})$

$$x_{k+1} = m^\dagger(x_k, \Psi_0(x_k));$$

we focus our experiments on the additive residual framing in (4.5).

Note that the discrete-time formulation can be made compatible with continuous-time data sampled uniformly at rate Δt (i.e. $x(k\Delta t) = x_k$ for $k \in \mathbb{N}$). To see this, let $\Phi^\dagger(x_0, t) := x(t)$ be the solution operator for (4.1) (and Φ_0 defined analogously for f_0). We then have

$$\Psi^\dagger(v) := \Phi^\dagger(v, \Delta t) \tag{4.6a}$$

$$\Psi_0(v) := \Phi_0(v, \Delta t), \tag{4.6b}$$

which can be obtained via numerical integration of $f^\dagger$, f_0, respectively.

4.2.3 Partially Observed Systems (Continuous-Time)

The framework in Sections 4.2.1 and 4.2.2 assumes that the system dynamics are Markovian with respect to observable x. Most of our experiments are performed in the fully-observed Markovian case. However, this assumption rarely holds in real-world systems. Consider a block-on-a-spring experiment conducted in an introductory physics laboratory. In principle, the system is strictly governed by the position and momentum of the block (i.e. f_0), along with a few scalar parameters. However (as most students' error analysis reports will note), the dynamics are also driven by a variety of external factors, like a wobbly table or a poorly greased track. The magnitude, timescale, and structure of the influence of these different factors are rarely known; and yet, they are somehow encoded in the discrepancy between the nominal equations of motion and the (noisy) observations of this multiscale system.

Thus we also consider the setting in which the dynamics of x is not Markovian. If we consider x to be the observable states of a Markovian system in dimension higher than d_x, then we can write the full system as

$$\dot{x} = f^\dagger(x, y), \quad x(0) = x_0 \tag{4.7a}$$

$$\dot{y} = \frac{1}{\varepsilon} g^\dagger(x, y), \quad y(0) = y_0. \tag{4.7b}$$

Here $f^\dagger \in C^1(\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}; \mathbb{R}^{d_x})$, $g^\dagger \in C^1(\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}; \mathbb{R}^{d_y})$, and $\varepsilon > 0$ is a constant measuring the degree of scale-separation (which is large when ε is small). The system yields solution [2] $x(\cdot) \in \mathsf{X}_T, y(\cdot) \in \mathsf{Y}_T$ for any $T < T_{\max}(x(0), y(0)) \in \mathbb{R}^+$, the maximal interval of existence. We view y as the complicated, unresolved, or unobserved aspects of the true underlying system.

For any $f_0 \in C^1(\mathbb{R}^{d_x}; \mathbb{R}^{d_x})$ (regardless of its fidelity), there exists a function $m^\dagger(x, y) \in C^1(\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}; \mathbb{R}^{d_x})$ such that (4.7) can be rewritten as

$$\dot{x} = f_0(x) + m^\dagger(x, y) \tag{4.8a}$$

$$\dot{y} = \frac{1}{\varepsilon} g^\dagger(x, y). \tag{4.8b}$$

Now observe that, by considering the solution of equation (4.8b) as a function of the history of x, the influence of $y(\cdot) \in \mathsf{Y}_t$ on the solution $x(\cdot) \in \mathsf{X}_t$ can be captured by a parameterized (w.r.t. t) family of operators $m_t^\dagger : \mathsf{X}_t \times \mathbb{R}^{d_y} \times \mathbb{R}^+ \mapsto \mathbb{R}^{d_x}$ on the historical trajectory $\{x(s)\}_{s=0}^t$, unobserved initial condition $y(0)$, and scale-separation parameter ε such that

$$\dot{x}(t) = f_0\big(x(t)\big) + m_t^\dagger\big(\{x(s)\}_{s=0}^t;\ y(0),\ \varepsilon\big). \tag{4.9}$$

Our goal is to use machine learning to find a Markovian model, in which x is part of the state variable, using our nominal knowledge f_0 and observations of a trajectory $\{x(t)\}_{t=0}^T \in \mathsf{X}_T$, for some $T < T_{\max}(x_0, y_0)$, from the true system (4.7); note that $y(\cdot)$ is not observed and nothing is assumed known about the vector field $g^\dagger$ or the parameter ε.

Note that equations (4.7), (4.8) and (4.9) are all equivalent formulations of the same problem and have identical solutions. The third formulation points towards two intrinsic difficulties: the unknown "function" to be learned is in fact defined by a family of operators $m_t^\dagger$ mapping the Banach space of path history into $\mathbb{R}^{d_x}$; secondly the operator is parameterized by $y(0)$ which is unobserved. We will address the first issue by showing that the operators $m_t^\dagger$ can be arbitrarily well-approximated from within a family of differential equations in dimension $\mathbb{R}^{2d_x + d_y}$; the second issue may be addressed by techniques from data assimilation [21, 227, 346] once this approximating family is learned. We emphasize, however, that we do not investigate the practicality of this approach to learning non-Markovian systems and much remains to be done in this area.

[2] With Y_T defined analogously to X_T.

It is also important to note that these non-Markovian operators $m_t^\dagger$ can sometimes be adequately approximated by invoking a Markovian model for x and simply learning function $m^\dagger(\cdot)$ as in Section 4.2.1. For example, when $\varepsilon \to 0$ and the y dynamics, with x fixed, are sufficiently mixing, the averaging principle [32, 415, 319] may be invoked to deduce that

$$\lim_{\varepsilon \to 0} m_t^\dagger\big(\{x(s)\}_{s=0}^t;\ y(0), \varepsilon\big) = m^\dagger(x(t))$$

for some $m^\dagger$ as in Section 4.2.1. This fact is used in section 3 of [193] to study the learning of closure models for linear Gaussian stochastic differential equations (SDEs).

It is highly advantageous to identify settings where Markovian modeling is sufficient, as it is a simpler learning problem. We find that learning $m_t^\dagger$ is necessary when there is significant memory required to explain the dynamics of x; learning $m^\dagger$ is sufficient when memory effects are minimal. In Section 4.6, we show that Markovian closures can perform well for certain tasks even when the scale-separation factor ε is not small. In Section 4.3 we demonstrate how the family of operators $m_t^\dagger$ may be represented through ODEs, appealing to ideas which blend continuous-time RNNs with an assumed known vector field f_0.

4.2.4 Partially Observed Systems (Discrete-Time)

The discrete-time analog of the previous setting considers a mapping

$$x_{k+1} = \Psi_1^\dagger(x_k, y_k) \tag{4.10a}$$
$$y_{k+1} = \Psi_2^\dagger(x_k, y_k) \tag{4.10b}$$

with $\Psi_1^\dagger \in C(\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}; \mathbb{R}^{d_x})$, $\Psi_2^\dagger \in C(\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}; \mathbb{R}^{d_y})$, yielding solutions $\{x_k\}_{k\in\mathbb{Z}^+} \in \mathsf{X}_\infty$ and $\{y_k\}_{k\in\mathbb{Z}^+} \in \mathsf{Y}_\infty$. We assume unknown $\Psi_1^\dagger, \Psi_2^\dagger$, but known approximate model Ψ_0 to rewrite (4.10) as

$$x_{k+1} = \Psi_0(x_k) + m^\dagger(x_k, y_k) \tag{4.11a}$$
$$y_{k+1} = \Psi_2^\dagger(x_k, y_k). \tag{4.11b}$$

We can, analogously to (4.9), write a solution in the space of observables as

$$x_{k+1} = \Psi_0\big(x_k\big) + m_k^\dagger\big(\{x_s\}_{s=0}^k,\ y_0\big) \tag{4.12}$$

with $m_k^\dagger\colon \mathsf{X}_k \times \mathbb{R}^{d_y} \longrightarrow \mathbb{R}^{d_x}$, a function of the historical trajectory $\{x_s\}_{s=0}^k$ and the unobserved initial condition y_0. If this discrete-time system is computed from the time Δt map for (4.1) then, for $\varepsilon \ll 1$ and when averaging scenarios apply as discussed in Section 4.2.3, the memoryless model in (4.5) may be used.

4.3 Statistical Learning for Ergodic Dynamical Systems

Here, we present a learning theory framework within which to consider methods for discovering model error from data. We outline the learning theory in a continuous-time Markovian setting (using possibly discretely sampled data), and point to its analogs in discrete-time and non-Markovian settings.

In the discrete-time settings, we assume access to discretely sampled training data $\{x_k = x(k\Delta t)\}_{k=0}^K$, where Δt is a uniform sampling rate and we assume that $K\Delta t = T$. In the continuous-time settings, we assume access to continuous-time training data $\{\dot{x}(t), x(t)\}_{t=0}^T$; Section 4.6.2.1 discusses the important practical question of estimating $\dot{x}(t), x(t)$ from discrete (but high frequency) data. In either case, consider the problem of identifying $m \in \mathcal{M}$ (where $\mathcal{M}$ represents the model, or hypothesis, class) that minimizes a loss function quantifying closeness of m to $m^\dagger$. In the Markovian setting we choose a measure μ on $\mathbb{R}^{d_x}$ and define the loss

$$\mathcal{L}_\mu(m, m^\dagger) := \int_{\mathbb{R}^{d_x}} \|m(x) - m^\dagger(x)\|_2^2 d\mu(x).$$

If we assume that, at the true $m^\dagger$, $x(\cdot)$ is ergodic with invariant density μ, then we can exchange time and space averages to see, for infinitely long trajectory $\{x(t)\}_{t\geq 0}$,

$$\begin{aligned}\mathcal{I}_\infty(m) &:= \lim_{T\to\infty} \frac{1}{T}\int_0^T \|m(x(t)) - m^\dagger(x(t))\|_2^2 dt\\ &= \int_{\mathbb{R}^{d_x}} \|m(x) - m^\dagger(x)\|_2^2 d\mu(x)\\ &= \mathcal{L}_\mu(m, m^\dagger).\end{aligned}$$

Since we may only have access to a trajectory dataset of finite length T, it is natural to define

$$\mathcal{I}_T(m) := \frac{1}{T}\int_0^T \|m(x(t)) - m^\dagger(x(t))\|_2^2 dt$$

and note that, by ergodicity,

$$\lim_{T\to\infty} \mathcal{I}_T(m) = \mathcal{L}_\mu(m, m^\dagger).$$

Finally, we can use (4.2) to get

$$\mathcal{I}_T(m) = \frac{1}{T}\int_0^T \|\dot{x}(t) - f_0(x(t)) - m(x(t))\|_2^2 dt. \tag{4.13}$$

This, possibly regularized, is a natural loss function to employ when continuous-time data is available, and should be viewed as approximating $\mathcal{L}_\mu(m, m^\dagger)$. We can use these definitions to frame the problem of learning model error in the language of statistical learning [416].

If we let $\mathcal{M}$ denote the hypothesis class over which we seek to minimize $\mathcal{I}_T(m)$ then we may define

$$m^*_\infty = \operatorname*{arg\,min}_{m\in\mathcal{M}} \mathcal{L}_\mu(m, m^\dagger) = \operatorname*{arg\,min}_{m\in\mathcal{M}} \mathcal{I}_\infty(m), \quad m^*_T = \operatorname*{arg\,min}_{m\in\mathcal{M}} \mathcal{I}_T(m).$$

The *risk* associated with seeking to approximate $m^\dagger$ from the class $\mathcal{M}$ is defined by $\mathcal{L}_\mu(m^*_\infty, m^\dagger)$, noting that this is 0 if $m^\dagger \in \mathcal{M}$. The risk measures the intrinsic error incurred by seeking to learn $m^\dagger$ from the restricted class $\mathcal{M}$, which typically does not include $m^\dagger$; it is an approximation theoretic concept which encodes the richness of the hypothesis class $\mathcal{M}$. The risk may be decreased by increasing the expressiveness of $\mathcal{M}$. Thus risk is independent of the data employed. *Empirical risk minimization* refers to minimizing $\mathcal{I}_T$ (or a regularized version) rather than $\mathcal{I}_\infty$, and this involves the specific instance of data that is available. To quantify the effect of data volume on learning $m^\dagger$ through empirical risk minimization, it is helpful to introduce the following two concepts. The *excess risk* is defined by

$$R_T := \mathcal{I}_\infty(m^*_T) - \mathcal{I}_\infty(m^*_\infty) \tag{4.14}$$

and represents the additional approximation error incurred by using data defined over a finite time horizon T in the estimate of $m^\dagger$. The *generalization error* is

$$G_T := \mathcal{I}_T(m^*_T) - \mathcal{I}_\infty(m^*_T) \tag{4.15}$$

and represents the discrepancy between training error, which is defined using a finite trajectory, and idealized test error, which is defined using an infinite length trajectory (or, equivalently, the invariant measure μ), when evaluated at the estimate of the function $m^\dagger$ obtained from finite data. We return to study excess risk and generalization error in the context of linear (in terms of parametric-dependence) models for $m^\dagger$, and under ergodicity assumptions on the data generating process, in Section 4.5.2.

We have introduced a machine learning framework in the continuous-time Markovian setting, but it may be adopted in discrete-time and in non-Markovian settings. In Section 4.4, we define appropriate objective functions for each of these cases.

Remark 4.3.1. The developments we describe here for learning in ODEs can be extended to the case of learning SDEs; see [33, 221]. In that setting, consistency in the large T limit is well-understood. It would be interesting to build on the learning theory perspective described here to study statistical consistency for ODEs; the approaches developed in the work by McGoff et al. [284] and Su and Mukherjee [400] are potentially useful in this regard. □

4.4 Parameterization of the Loss Function

In this section, we define explicit optimizations for learning (approximate) model error functions $m^\dagger$ for the Markovian settings, and model error operators $m_t^\dagger$ for the non-Markovian settings; both continuous- and discrete-time formulations are given. We defer discussion of specific approximation architectures to the next section. Here we make a notational transition from optimization over (possibly non-parametric) functions $m \in \mathcal{M}$ to functions parameterized by θ that characterize the class $\mathcal{M}$.

In all the numerical experiments in this paper, we study the use of continuous- and discrete-time approaches to model data generated by a continuous-time process. The set-up in this section reflects this setting, in which two key parameters appear: T, the continuous-time horizon for the data; and Δt, the frequency of the data. The latter parameter will always appear in the discrete-time models; but it may also be implicit in continuous-time models which need to infer continuous-time quantities from discretely sampled data. We relate T and Δt by $K\Delta t = T$. We present the general forms of $\mathcal{J}_T(\theta)$ (with optional regularization terms $R(\theta)$). Optimization via derivative-based methodology requires either analytic differentiation of the dynamical system model with respect to parameters, or the use of autodifferentiable ODE solvers [358].

4.4.1 Continuous-Time Markovian Learning

Here, we approximate the Markovian closure term in (4.2) with a parameterized function $m(x;\ \theta)$. Assuming full knowledge of $\dot{x}(t), x(t)$, we learn the correction term for the flow field by minimizing the following objective function of θ:

$$\mathcal{J}_T(\theta) = \frac{1}{T}\int_0^T \left\|\dot{x}(t) - f_0(x(t)) - m(x(t);\ \theta)\right\|^2 dt + R(\theta) \tag{4.16}$$

Note that $\mathcal{J}_T(\theta) = \mathcal{I}_T\big(m(\,\cdot\,;\,\theta)\big) + R(\theta)$; thus the proposed methodology is a regularization of the empirical risk minimization described in the preceding section.

Notable examples that leverage this framing include: the paper [196], where θ are coefficients for a library of low-order polynomials and $R(\theta)$ is a sparsity-promoting regularization defined by the SINDy framework; the paper [446], where θ are parameters of a deep neural network (DNN) and L_2 regularization is applied to the weights; the paper [382], where θ are DNN parameters and $R(\theta)$ encodes constraints on the Lipschitz constant for m provided by spectral normalization; and the paper [429] which applies this approach to the Lorenz '96 Multiscale system using neural networks with an L_2 regularization on the weights.

4.4.2 Discrete-Time Markovian Learning

Here, we learn the Markovian correction term in (4.5) by minimizing:

$$\mathcal{J}_T(\theta) = \frac{1}{K}\sum_{k=0}^{K-1}\Big\|x_{k+1} - \Psi_0(x_k) - m(x_k;\,\theta)\Big\|^2 + R(\theta) \tag{4.17}$$

This is the natural discrete-time analog of (4.16) and may be derived analogously, starting from a discrete analog of the loss $\mathcal{L}_\mu(m, m^\dagger)$ where now μ is assumed to be an ergodic measure for (4.4). If a discrete analog of (4.13) is defined, then parameterization of m, and regularization, leads to (4.17). This is the underlying model assumption in the work by Farchi et al. [131].

4.4.3 Continuous-Time Non-Markovian Learning

We can attempt to recreate the dynamics in x for (4.9) by modeling the non-Markovian residual term. A common approach is to augment the dynamics space with a variable $r \in \mathbb{R}^{d_r}$ leading to a model of the form

$$\dot{x} = f_0(x) + f_1(r, x;\,\theta) \tag{4.18a}$$

$$\dot{r} = f_2(r, x;\,\theta). \tag{4.18b}$$

We then seek a d_r large enough, and then parametric models $\{f_j(r, x; \cdot)\}_{j=1}^2$ expressive enough, to ensure that the dynamics in x are reproduced by (4.18). Note that, although the model error in x is non-Markovian, as it depends on the history of x, we are seeking to explain observed x data by an enlarged model, including hidden variables r, in which the dynamics of $[x, r]$ is Markovian.

When learning hidden dynamics from partial observations, we must jointly infer the missing states $r(t)$ and the, typically parameterized, governing dynamics f_1, f_2. Furthermore, when the family of parametric models is not closed with respect to translation of r it will also be desirable to learn r_0; when x is observed noisily, it is similarly important to learn x_0.

To clarify discussions of (4.18) and its training from data, let $u = [x, r]$ and f be the concatenation of the vector fields given by f_0, f_1, f_2 such that

$$\dot{u} = f(u;\ \theta), \tag{4.19}$$

with solution $u(t;\ v, \theta)$ solving (4.19) (and, equivalently, (4.18)) with initial condition v (i.e. $u(0;\ v, \theta) = v$). Consider observation operators H_x, H_r, such that $x = H_x u$, and $r = H_r u$, and further define noisy observations of x as

$$z(t) = x(t) + \eta(t),$$

where η is i.i.d. observational noise. We now outline three optimization approaches to learning from noisily, partially observed data z.

4.4.3.1 Optimization; Hard Constraint For Missing Dynamics

Since (4.18) is deterministic, it may suffice to jointly learn parameters and initial condition $u(0) = u_0$ by minimizing [358]:

$$\mathcal{J}_T(\theta, u_0) = \frac{1}{T}\int_0^T \left\| z(t) - H_x u(t;\ u_0, \theta) \right\|^2 dt + R(\theta) \tag{4.20}$$

A similar approach was applied in [24], but where initial conditions were learnt as outputs of an additional DNN encoder network that maps observation sequences (of fixed length and temporal discretization) to initial conditions.

4.4.3.2 Optimization; Weak Constraint For Missing Dynamics

The hard constraint minimization is very sensitive for large T in settings where the dynamics is chaotic. This can be ameliorated, to some extent, by considering the objective function

$$\begin{aligned} \mathcal{J}_T\big(\theta, u(t)\big) = &\frac{1}{T}\int_0^T \left\| z(t) - H_x u(t) \right\|^2 dt \\ &+ \lambda \frac{1}{T}\int_0^T \left\| \dot{u}(t) - f(u(t);\ \theta) \right\|^2 dt. \end{aligned} \tag{4.21}$$

This objective function is employed in [304], where it is motivated by the weak-constraint variational formulation (4DVAR) arising in data assimilation [227].

4.4.3.3 Optimization; Data Assimilation For Missing Dynamics

The weak constraint approach may still scale poorly with T large, and still relies on gradient-based optimization to infer hidden states. To avoid these potential issues, we follow the recent work of [82], using filtering-based methods to estimate the hidden state. This implicitly learns initializations and it removes noise from data. It allows computation of gradients of the resulting loss function back through the filtering algorithm to learn model parameters. We define a filtered state

$$\hat{u}_{t,\tau} := \hat{u}_t\Big(\tau;\ \hat{v}, \theta_{\mathrm{DYN}}, \theta_{\mathrm{DA}}, \big\{z(t+s)\big\}_{s=0}^{\tau}\Big)$$

as an estimate of $u(t+\tau)|\{z(t+s)\}_{s=0}^{\tau}$ when initialized at $\hat{u}_{t,0} = \hat{v}$. [3] In this formulation, we distinguish θ_{DYN} as parameters for modeling dynamics via (4.18), and θ_{DA} as hyper-parameters governing the specifics of a data assimilation scheme. Examples of θ_{DA} are the constant gain matrix K that must be chosen for 3DVAR, or parameters of the inflation and localization methods deployed within Ensemble Kalman Filtering. By parameterizing these choices as θ_{DA}, we can optimize them jointly with model parameters θ_{DYN}. To do this, let $\theta = [\theta_{\mathrm{DYN}}, \theta_{\mathrm{DA}}]$ and minimize

$$\mathcal{J}_T(\theta) = \frac{1}{(T-\tau_1-\tau_2)\tau_2} \int_{t=0}^{T-\tau_1-\tau_2} \int_{s=0}^{\tau_2} \Big\| z(t+\tau_1+s) - H_x u(s;\ \hat{u}_{t,\tau_1}, \theta) \Big\|^2 ds\, dt. \tag{4.22}$$

Here, τ_1 denotes the length of assimilation time used to estimate the state which initializes a parameter-fitting over window of duration τ_2; this parameter-fitting leads to the inner-integration over s. This entire picture is then translated through t time units and the objective function is found by integrating over t. Optimizing (4.22) can be understood as a minimization over short-term forecast errors generated from all assimilation windows. The inner integral takes a fixed start time t, applies data assimilation over a window $[t, t+\tau_1]$ to estimate an initial condition $\hat{u}_{t,\tau_1}$, then computes a short-term (τ_2) prediction error resulting from this DA-based initialization. The outer integral sums these errors over all available windows in long trajectory of data of length T.

In our work, we perform filtering using a simple 3DVAR method, whose constant gain can either be chosen as constant, or can be learnt from data. When constant, a natural choice is $K \propto H_x^T$, and this approach has a direct equivalence to standard warmup strategies employed in RNN and RC training [421, 316]. The paper [82] suggests minimization of a similar objective, but considers more general observation

[3] In practice we have found that setting $\hat{v} = 0$ works well.

operators h, restricts the outer integral to non-overlapping windows, and solves the filtering problem with an EnKF with known state-covariance structure.

Remark 4.4.1. To motivate learning parameters of the data assimilation we make the following observation: for problems in which the model is known (i.e. θ_{DYN} is fixed) we observe successes with the approach of identifying 3DVAR gains that empirically outperform the theoretically derived gains in [224]. Similar is to be expected for parameters defining inflation and localization in the EnKF. □

Remark 4.4.2. Specific functional forms of f_1, f_2 (and their corresponding parameter inference strategies) reduce (4.18) to various approaches. For the continuous-time RNN analysis that we discuss in Section 4.5 we will start by considering settings in which f_1 and f_2 are approximated from expressive function classes, such as neural networks. We will then specify to models in which f_1 is linear in r and independent of x, whilst f_2 is a single layer neural network. It is intuitive that the former may be more expressive and allow a smaller d_r than the latter; but the latter connects directly to reservoir computing, a connection we make explicitly in what follows. Our numerical experiments in Section 4.6 will be performed in both settings: we will train models from the more general setting; and by carefully designed experiments we will shed light on issues arising from over-parameterization, in the sense of choosing to learn a model in dimension higher than that of the true observed-hidden model, working in the setting of linear coupling term f_1, depending only on r. □

Remark 4.4.3. The recent paper [157] proposes an interesting, and more computationally tractable, approach to learning model error in the presence of memory. They propose to learn a closure operator $m_\tau(\cdot\,;\,\theta) : \mathsf{X}_\tau \longrightarrow \mathbb{R}^{d_x}$ for a DDE with finite memory τ:

$$\dot{x}(t) = f_0\big(x(t)\big) + m_\tau\big(\{x(t-s)\}_{s=0}^{\tau};\ \theta\big); \tag{4.23}$$

neural networks are used to learn the operator m_τ. Alternatively, Gaussian processes are used to fit a specific class of stochastic delay differential equation (SDDE) (4.23) in [375]. However, although delay-based approaches have seen some practical success, in many applications they present issues for domain interpretability and Markovian ODE or PDE closures are more desirable. □

4.4.4 Discrete-Time Non-Markovian Learning

In similar spirit to Section 4.4.3, we can aim to recreate discrete-time dynamics in x for (4.12) with model

$$x_{k+1} = \Psi_0(x_k) + \Psi_1(r_k, x_k;\ \theta) \tag{4.24a}$$

$$r_{k+1} = \Psi_2(r_k, x_k;\ \theta) \tag{4.24b}$$

and objective function

$$\begin{aligned} \mathcal{J}_T(\theta, r_0) &= \frac{1}{K}\sum_{k=0}^{K-1} \left\| x_{k+1} - \Psi_0(x_k) - \Psi_1(r_k, x_k;\ \theta) \right\|^2 + R(\theta) \\ \text{s.t.} \quad & \{r_k\}_{k=1}^{K-1} \quad \text{solves (4.24b).} \end{aligned} \tag{4.25}$$

Observe that estimation of initial condition r_0 is again crucial, and the data assimilation methods discussed in Section 4.4.3 can be adapted to this discrete-time setting. The functional form of Ψ_1, Ψ_2 (and their corresponding parameter inference strategies) reduce (4.24) to various approaches, including recurrent neural networks, latent ODEs, and delay-embedding maps (e.g. to get a delay embedding map, Ψ_2 is a shift operator). Pathak et al. [316] use reservoir computing (a random features analog to RNN, described in the next section) with L_2 regularization to study an approach similar to (4.24), but included $\Psi_0(x_k)$ as a feature in Ψ_1 and Ψ_2 instead of using it as the central model upon which to learn residuals. The data-driven superparameterization approach in [75] also appears to follow the underlying assumption of (4.24). Harlim et al. [165] evaluate hybrid models of form (4.24) both in settings where delay embedding closures are employed and where RNN-based approximations via LSTMs are employed.

4.5 Underpinning Theory

In this section we identify specific hypothesis classes $\mathcal{M}$. We do this using random feature maps [335] in the Markovian settings (Section 4.5.1), and using recurrent neural networks in the memory-dependent setting. We then discuss these problems from a theoretical standpoint. In Section 4.5.2 we study excess risk and generalization error in the context of linear models (a setting which includes the random features model as a special case). And we conclude by discussing the use of RNNs [149][Chapter 10] for the non-Markovian settings (discrete- and continuous-time) in Section 4.5.3; we present an approximation theorem for continuous-time hybrid RNN models. Throughout this section, the specific use of random feature maps and of recurrent neural networks is for illustration only; other models could, of course, be used.

4.5.1 Markovian Modeling with Random Feature Maps

In principle, any hypothesis class can be used to learn $m^\dagger$. However, we focus on models that are easily trained on large-scale complex systems and yet have proven approximation power for functions between finite-dimensional Euclidean spaces. For the Markovian modeling case, we use random feature maps; like traditional neural networks, they possess arbitrary approximation power [338, 337], but further benefit from a quadratic minimization problem in the training phase, as do kernel or Gaussian process methods. In our case studies, we found random feature models sufficiently expressive, we found optimization easily implementable, and we found the learned models generalized well. Moreover, their linearity with respect to unknown parameters enables a straightforward analysis of excess risk and generalization error in Section 4.5.2. Details on the derivation and specific design choices for our random feature modeling approach can be found in Section 4.8.4, where we explain how we sample D random feature functions $\varphi : \mathbb{R}^{d_x} \to \mathbb{R}$ and stack them to form a vector-valued feature map $\phi \colon \mathbb{R}^{d_x} \to \mathbb{R}^D$. Given this random function ϕ, we define the hypothesis class

$$\mathcal{M} = \{m \colon \mathbb{R}^{d_x} \to \mathbb{R}^{d_x} \mid \exists\, C \in \mathbb{R}^{d_x \times D} : m(x) = C\phi(x)\}. \tag{4.26}$$

4.5.1.1 Continuous-Time

In the continuous-time framing, our Markovian closure model uses hypothesis class (4.26) and thus takes the form

$$\dot{x} = f_0(x) + C\phi(x(t)).$$

We rewrite (4.16) for this particular case with an L_2 regularization parameter $\lambda \in \mathbb{R}^+$:

$$\mathcal{J}_T(C) = \frac{1}{2T}\int_0^T \left\|\dot{x}(t) - f_0(x(t)) - C\phi\big(x(t)\big)\right\|^2 dt + \frac{\lambda}{2}\|C\|^2. \tag{4.27}$$

We employ the notation $A \otimes B := AB^T$ for the outer-product between matrices $A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{l\times n}$, and the following notation for time-average

$$\overline{A}_T := \frac{1}{T}\int_0^T A(t)dt$$

of $A \in L^1([0,T],\mathbb{R}^{m\times n})$. The objective function in (4.27) is quadratic and convex in C and thus is globally minimized for the unique C^* which makes the derivative of $\mathcal{J}_T$ zero. Consequently, the minimizer C^* satisfies the following linear equation (derived in Section 4.8.5):

$$(Z+\lambda I)(C^*)^T = Y. \tag{4.28}$$

Here, $I \in \mathbb{R}^{D\times D}$ is the identity and

$$\begin{aligned} Z &= \overline{[\phi\otimes\phi]}_T \in \mathbb{R}^{D\times D},\\ Y &= \overline{[\phi\otimes m^\dagger]}_T \in \mathbb{R}^{D\times d_x}. \end{aligned} \tag{4.29}$$

Of course $m^\dagger$ is not known, but $m^\dagger(t) = \dot{x}(t) - f_0(x(t))$ can be computed from data.

To summarize, the algorithm proceeds as follows: 1) create a realization of random feature vector ϕ; 2) compute the integrals in (4.29) to obtain Z, Y; and 3) solve the linear matrix equation (4.28) for C^*. Together this leads to our approximation $m^\dagger(x) \approx m_T^*(x;\,\theta) := C^*\phi(x)$.

4.5.1.2 Discrete-Time

In discrete-time, our Markovian closure model is

$$x_{k+1} = \Psi_0(x_k) + C\phi(x_k),$$

and is learnt by minimizing

$$\mathcal{J}_T(\theta) = \frac{1}{K}\sum_{k=0}^{K-1}\left\|x_{k+1} - \Psi_0(x_k) - C\phi\big(x(t)\big)\right\|^2 + \frac{\lambda}{2}\|C\|^2. \tag{4.30}$$

The objective function in (4.30) is quadratic in C and thus globally minimized at C^*. As in Section 4.5.1.1, we can compute Z, Y and solve a linear system for C^* to approximate $m^\dagger(x) \approx m_T^*(x;\,\theta) := C^*\phi(x)$. This formulation closely mirrors the fully data-driven linear regression approach in [151].

4.5.2 Learning Theory for Markovian Models with Linear Hypothesis Class

In this subsection we provide estimates of the excess risk and generalization error in the context of learning $m^\dagger$ in (4.2) from a trajectory over time horizon T. We study ergodic continuous-time models in the setting of Section 4.4.1. To this end we consider the very general linear hypothesis class given by

$$\mathcal{M} = \{m\colon \mathbb{R}^{d_x} \to \mathbb{R}^{d_x} \mid \exists\, \theta \in \mathbb{R}^p : m(x) = \sum_{\ell=1}^{p}\theta_\ell f_\ell(x)\}; \tag{4.31}$$

we note that if the $\{f_\ell\}$ are i.i.d. draws of function ϕ in the case $D = d_x$ then this too reduces to a random features model, but that our analysis in the context of statistical learning does not rely on the random features structure. In fact our analysis can be used to provide learning theory for other linear settings, where $\{f_\ell\}$ represents a dictionary of hypothesized features whose coefficients are to be learnt from data. Nonetheless, universal approximation for random features [335] provides an important example of an approximation class for which the loss function $\mathcal{I}_\infty$ may be made arbitrarily small by choice of p large enough and appropriate choice of parameters, and the reader may find it useful to focus on this case. We also note that the theory we present in this subsection is readily generalized to working with hypothesis class (4.26).

We make the following ergodicity assumption about the data generation process:

Assumption 4.5.1. *Equation* (4.2) *possesses a compact attractor* $\mathcal{A}$ *supporting invariant measure* μ*. Furthermore the dynamical system on* $\mathcal{A}$ *is ergodic with respect to* μ *and satisfies a central limit theorem of the following form: for all Hölder continuous* $\varphi : \mathbb{R}^{d_x} \mapsto \mathbb{R}$*, there is* $\sigma^2 = \sigma^2(\varphi)$ *such that*

$$\sqrt{T}\left(\frac{1}{T}\int_0^T \varphi\big(x(t)\big)dt - \int_{\mathbb{R}^{d_x}} \varphi\big(x\big)\mu(dx)\right) \Rightarrow N(0,\sigma^2) \tag{4.32}$$

where $\Rightarrow$ *denotes convergence in distribution with respect to* $x(0) \sim \mu$*. Furthermore a law of the iterated logarithm holds: almost surely with respect to* $x(0) \sim \mu$*,*

$$limsup_{T\to\infty}\left(\frac{T}{\log\log T}\right)^{\frac{1}{2}}\left(\frac{1}{T}\int_0^T \varphi\big(x(t)\big)dt - \int_{\mathbb{R}^{d_x}} \varphi\big(x\big)\mu(dx)\right) = \sigma. \tag{4.33}$$

Remark 4.5.2. Note that in both (4.32) and (4.33) $\varphi(\cdot)$ is only evaluated on (compact) $\mathcal{A}$ obviating the need for any boundedness assumptions on $\varphi(\cdot)$. In the work of Melbourne and co-workers, Assumption 4.5.1 is proven to hold for a class of differential equations, including the Lorenz '63 model at, and in a neighbourhood of, the classical parameter values: in [178] the central limit theorem is established; and in [27] the continuity of σ in φ is proven. Whilst it is in general very difficult to prove such results for any given chaotic dynamical system, there is strong empirical evidence for such results in many chaotic dynamical systems that arise in practice. This combination of theory and empirical evidence justify studying the learning of model error under Assumption 4.5.1. Tran and Ward [408] were the first to make use of the theory of Melbourne and coworkers to study learning of chaotic differential equations from time-series. □

Given m from hypothesis class $\mathcal{M}$ defined by (4.31) we define

$$\theta^*_\infty = \operatorname*{arg\,min}_{\theta\in\mathbb{R}^p} \mathcal{I}_\infty\big(m(\cdot\,;\theta)\big) = \operatorname*{arg\,min}_{\theta\in\mathbb{R}^p} \mathcal{L}_\mu\big(m(\cdot\,;\theta)\big) \tag{4.34}$$

and

$$\theta^*_T = \operatorname*{arg\,min}_{\theta\in\mathbb{R}^p} \mathcal{I}_T\big(m(\cdot\,;\theta)\big). \tag{4.35}$$

(Regularization is not needed in this setting because the data is plentiful—a continuous-time trajectory—and the number of parameters is finite.) Then $\theta^*_\infty, \theta^*_T$ solve the linear systems

$$A_\infty\theta^*_\infty = b_\infty, \quad A_T\theta^*_T = b_T$$

where

$$(A_\infty)_{ij} = \int_{\mathbb{R}^{d_x}} \big\langle f_i(x), f_j(x)\big\rangle\, \mu(dx), \qquad (b_\infty)_j = \int_{\mathbb{R}^{d_x}} \big\langle m^\dagger(x), f_j(x)\big\rangle\, \mu(dx),$$
$$(A_T)_{ij} = \frac{1}{T}\int_0^T \big\langle f_i\big(x(t)\big), f_j\big(x(t)\big)\big\rangle\, dt, \qquad (b_T)_j = \frac{1}{T}\int_0^T \big\langle m^\dagger\big(x(t)\big), f_j\big(x(t)\big)\big\rangle\, dt.$$

These facts can be derived analogously to the derivation in Section 4.8.5. Given θ^*_∞ and θ^*_T we also define

$$m^*_\infty = m(\cdot\,;\theta^*_\infty), \;\; m^*_T = m(\cdot\,;\theta^*_T).$$

Recall that it is assumed that $f^\dagger$, f_0, and $m^\dagger$ are C^1. We make the following assumption regarding the vector fields defining hypothesis class $\mathcal{M}$.

Assumption 4.5.3. *The functions $\{f_\ell\}_{\ell=0}^p$ appearing in definition (4.31) of the hypothesis class $\mathcal{M}$ are Hölder continuous on $\mathbb{R}^{d_x}$. In addition, the matrix A_∞ is invertible.*

Theorem 4.5.4. *Let Assumptions 4.5.1 and 4.5.3 hold. Then the scaled excess risk $\sqrt{T}R_T$ in (4.14) (resp. scaled generalization error $\sqrt{T}|G_T|$ in (4.15)) is bounded above by $\|\mathcal{E}_R\|$ (resp. $\|\mathcal{E}_G\|$), where random variable $\mathcal{E}_R \in \mathbb{R}^p$ (resp. $\mathcal{E}_G \in \mathbb{R}^{p+1}$) converges in distribution to $N(0,\Sigma_R)$ (resp. $N(0,\Sigma_G)$) w.r.t. $x(0)\sim\mu$ as $T\to\infty$. Furthermore, there is constant $C>0$ such that, almost surely w.r.t. $x(0)\sim\mu$,*

$$limsup_{T\to\infty}\Big(\frac{T}{\log\log T}\Big)^{\frac{1}{2}}\big(R_T + |G_T|\big) \le C.$$

The proof is provided in Section 4.8.1.

Remark 4.5.5. The convergence in distribution shows that, with high probability with respect to initial data, the excess risk and the generalization error are bounded above by terms of size $1/\sqrt{T}$. This can be improved to give an almost sure result, at the cost of the factor of $\sqrt{\log \log T}$. The theorem shows that (ignoring log factors and acknowledging the probabilistic nature of any such statements) trajectories of length $\mathcal{O}(\epsilon^{-2})$ are required to produce bounds on the excess risk and generalization error of size $\mathcal{O}(\epsilon)$.

The bounds on excess risk and generalization error also show that empirical risk minimization (of $\mathcal{I}_T$) approaches the theoretically analyzable concept of risk minimization (of $\mathcal{I}_\infty$) over hypothesis class (4.31). The sum of the excess risk R_T and the generalization error G_T gives

$$E_T := \mathcal{I}_T(m_T^*) - \mathcal{I}_\infty(m_\infty^*).$$

We note that $\mathcal{I}_T(m_T^*)$ is computable, once the approximate solution m_T^* has been identified; thus, when combined with an estimate for E_T, this leads to an estimate for the risk associated with the hypothesis class used.

If the approximating space $\mathcal{M}$ is rich enough, then approximation theory may be combined with Theorem 4.5.4 to estimate the trajectory error resulting from the learned dynamical system. Such an approach is pursued in Proposition 3 of [453] for SDEs. Furthermore, in that setting, knowledge of rate of mixing/decay of correlations for SDEs may be used to quantify constants appearing in the error bounds. It would be interesting to pursue such an analysis for chaotic ODEs with known mixing rates/decay of correlations. Such results on mixing are less well-developed, however, for chaotic ODEs; see discussion of this point in [178], and the recent work [27].

Work by Zhang, Harlim, and Li [452] demonstrates that error bounds on learned model error terms can be extended to bound error on reproduction of invariant statistics for ergodic SDEs. Moreover, E, Ma, and Wu [120] provide a direction for proving similar bounds on model error learning using nonlinear function classes (e.g. two-layer neural networks).

Finally we remark on the dependence of the risk and generalization error bounds on the size of the model error. It is intuitive that the amount of data required to learn model error should decrease as the size of the model error decreases. This is demonstrated numerically in Section 4.6.2.3 (c.f. Figures 4.2a and 4.2b). Here we comment that Theorem 4.5.4 also exhibits this feature: examination of the proof in Section 4.8.1 shows that all upper bounds on terms appearing in the excess and generalization error are proportional to $m^\dagger$ itself or to m^*_∞, its approximation given an infinite amount of data; note that $m^*_\infty = m^\dagger$ if the hypothesis class contains the truth. □

4.5.3 Non-Markovian Modeling with Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are one of the *de facto* tools for modeling systems with memory. Here, we show straightforward residual implementations of RNNs for continuous- and discrete-time, with the goal of modeling non-Markovian model error.

4.5.3.1 General Case

Equation (4.18b), and its coupling to (4.18a), constitute a very general way to account for memory-dependent model error in the dynamics of x. In fact, for f_1, f_2 sufficiently expressive (e.g. random feature functions, neural networks, polynomials), and $d_r \geq d_y$, solutions to (4.18) can approximate solutions to (4.8) arbitrarily well. We make this type of universal approximation theorem concrete in Theorems 4.5.10 and 4.5.15. We start by proving Theorem 4.5.10, which rests on the following assumptions:

Assumption 4.5.6. *Functions $f^\dagger, g^\dagger, f_0, f_1, f_2$ are all globally Lipschitz.*

Note that this implies that $m^\dagger$ is also globally Lipschitz.

Assumption 4.5.7. *Fix $T > 0$. There exist $\rho_0 \in \mathbb{R}, \rho_T \in \mathbb{R}$ such that, for equation (4.8), $\big(x(0), y(0)\big) \in B(0, \rho_0)$ implies that $\big(x(t), y(t)\big) \in B(0, \rho_T)\ \forall\ t \in [0, T]$.*

Assumption 4.5.8. *The hidden state in (4.18), $r \in \mathbb{R}^{d_r}$, has the same dimension as the true hidden state y in (4.8); that is $d_r = d_y$.*

Assumption 4.5.9. *Let functions $f_1(\cdot\ ;\ \theta) \in C^1(\mathbb{R}^{d_x} \times \mathbb{R}^{d_y};\ \mathbb{R}^{d_x})$ and $f_2(\cdot\ ;\ \theta) \in C^1(\mathbb{R}^{d_x} \times \mathbb{R}^{d_y};\ \mathbb{R}^{d_y})$ be parameterized [4] by $n \in \mathbb{N}$ and $\theta \in \mathbb{R}^n$. Then, for any $\delta > 0$, there exists $n > 0$ and $\theta \in \mathbb{R}^n$ such that*

$$\sup_{x,y \in B(0,\rho_T)} \|f^\dagger(x, y) - f_1(x, y;\ \theta)\| \leq \delta$$

and

$$\sup_{x,y \in B(0,\rho_T)} \|g^\dagger(x, y) - f_2(x, y;\ \theta)\| \leq \delta$$

[4] Here we define $\mathbb{N} = \{1, 2, \dots, \}$, the strictly positive integers.

Note that Theorem 4.5.9 can be satisfied by any parametric function class possessing a universal approximation property for maps between finite-dimensional Euclidean spaces, such as neural networks, polynomials and random feature methods. The next theorem transfers this universal approximation property for maps between Euclidean spaces to a universal approximation property for representation of model error with memory; this is a form of infinite dimensional approximation since, via its own dynamics, the memory variable r maps the past history of x into the model error correction term in the dynamics for x.

Theorem 4.5.10. *Let Assumptions 4.5.6-4.5.9 hold. Fix any $T > 0$ and $\rho_0 > 0$, let $x(\cdot), y(\cdot)$ denote the solution of (4.8) with $\varepsilon = 1$ and let $x_\delta(\cdot), r_\delta(\cdot)$ denote the solution of (4.18) with parameters $\theta \in \mathbb{R}^n$. Then, for any $\delta > 0$ and any $T > 0$, there is a parameter dimension $n = n_\delta \in \mathbb{N}$ and parameterization $\theta = \theta_\delta \in \mathbb{R}^{n_\delta}$ with the property that, for any initial condition $\big(x(0), y(0)\big) \in B(0, \rho_0)$ for (4.8), there is initial condition $(x_\delta(0), r_\delta(0)) \in \mathbb{R}^{d_x+d_y}$ for (4.18), such that*

$$\sup_{t\in[0,T]} \|x - x_\delta\| \leq \delta.$$

The proof is provided in Section 4.8.2; it is a direct consequence of the approximation power of f_1, f_2 and the Gronwall Lemma.

Remark 4.5.11. Note that this existence theorem also holds for $d_r > d_y$ by freezing the dynamics in the excess dimensions and initializing it at, for example, 0. However it is possible for augmentations with $d_r > d_y$ to introduce numerical instability when imperfectly initialized in the excess dimensions, despite their provable correctness when perfectly initialized (see Section 4.6.4). Nevertheless, we did not encounter such issues when training the general model class on the examples considered in this paper – see Section 4.6.3). □

4.5.3.2 Linear Coupling

We now study a particular form RNN in which the coupling term f_1 appearing in (4.18) is linear and depends only on the hidden variable:

$$\dot{x} = f_0(x) + Cr \tag{4.36a}$$

$$\dot{r} = \sigma(Ar + Bx + c). \tag{4.36b}$$

Here σ is an activation function. The specific linear coupling form is of particular interest because of the connection we make (see Remark 4.5.18 below) to reservoir computing. The goal is to choose A, B, C, c so that output $\{x(t)\}_{t\geq 0}$ matches output of (4.8), without observation of $\{y(t)\}_{t\geq 0}$ or knowledge of $m^\dagger$ and $g^\dagger$. As in the general case from the preceding subsection, inherent in choosing these matrices A, B, C and vector c is a choice of embedding dimension for variable r which will typically be larger than dimension of y itself. The idea is to create a recurrent state r of sufficiently large dimension d_r whose evolution equation takes x as input and, after a final linear transformation, approximates the missing dynamics $m^\dagger(x, y)$.

There is existing approximation theory for discrete-time RNNs [370] showing that a discrete-time analog of our linear coupling set-up can be used to approximate discrete-time systems arbitrarily well; see also Theorem 3 of [165]. There is also a general approximation theorem using continuous-time RNNs proved in [139], but it does not apply to the linear-coupling setting. We thus extend the work in these three papers to the context of residual-based learning as in (4.36). We state the theorem after making three assumptions upon which it rests:

Assumption 4.5.12. *Functions $f^\dagger, g^\dagger, f_0$ are all globally Lipschitz.*

Note that this implies that $m^\dagger$ is also globally Lipschitz.

Assumption 4.5.13. *Let $\sigma_0 \in C^1(\mathbb{R};\mathbb{R})$ be bounded and monotonic, with bounded first derivative. Then $\sigma(u)$ defined by $\sigma(u)_i = \sigma_0(u_i)$ satisfies $\sigma \in C^1(\mathbb{R}^p;\mathbb{R}^p)$.*

Assumption 4.5.14. *Fix $T > 0$. There exist $\rho_0 \in \mathbb{R}, \rho_T \in \mathbb{R}$ such that, for equation (4.8), $\big(x(0), y(0)\big) \in B(0, \rho_0)$ implies that $\big(x(t), y(t)\big) \in B(0, \rho_T)\ \forall\, t \in [0, T]$.*

Theorem 4.5.15. *Let Assumptions 4.5.12-4.5.14 hold. Fix any $T > 0$ and $\rho_0 > 0$, let $x(\cdot), y(\cdot)$ denote the solution of (4.8) with $\varepsilon = 1$ and let $x_\delta(\cdot), r_\delta(\cdot)$ denote the solution of (4.36) with parameters $\theta \in \mathbb{R}^n$. Then, for any $\delta > 0$ and any $T > 0$, there is embedding dimension $d_r \in \mathbb{N}$, parameter dimension $n = n_\delta \in \mathbb{N}$ and parameterization $\theta = \theta_\delta = \{A_\delta,\ B_\delta,\ C_\delta, c_\delta\}$ with the property that, for any initial condition $\big(x(0), y(0)\big) \in B(0, \rho_0)$ for (4.8), there is initial condition $(x_\delta(0), r_\delta(0)) \in \mathbb{R}^{d_x+d_r}$ for (4.36), such that*

$$\sup_{t\in[0,T]} \|x - x_\delta\| \leq \delta.$$

The complete proof is provided in Section 4.8.3; here we describe its basic structure. Define $m(t) := m^\dagger\big(x(t), y(t)\big)$ and, with the aim of finding a differential equation for $m(t)$, recall (4.8) with $\varepsilon = 1$ and define the vector field

$$h^\dagger(x, y) := \nabla_x m^\dagger(x, y)[f_0(x) + m^\dagger(x, y)] + \nabla_y m^\dagger(x, y) g^\dagger(x, y). \tag{4.37}$$

Since $\dot{m}(t)$ is the time derivative of $m^\dagger\big(x(t), y(t)\big)$, when (x, y) solve (4.8) we have

$$\dot{m} = h^\dagger(x, y).$$

Motivated by these observations, we now introduce a new system of autonomous ODEs for the variables $(x, y, m) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y} \times \mathbb{R}^{d_x}$:

$$\dot{x} = f_0(x) + m \tag{4.38a}$$
$$\dot{y} = g^\dagger(x, y) \tag{4.38b}$$
$$\dot{m} = h^\dagger(x, y). \tag{4.38c}$$

To avoid a proliferation of symbols we use the same letters for (x, y) solving equation (4.38) as for (x, y) solving equation (4.8). We now show $m = m^\dagger(x, y)$ is an invariant manifold for (4.38); clearly, on this manifold, the dynamics of (x, y) governed by (4.38) reduces to the dynamics of (x, y) governed by (4.8). Thus $m(t)$ must be initialized at $m^\dagger\big(x(0), y(0)\big)$ to ensure equivalence between the solution of (4.38) and (4.8).

The desired invariance of manifold $m = m^\dagger(x, y)$ under the dynamics (4.38) follows from the identity

$$\frac{d}{dt}\Big(m - m^\dagger(x, y)\Big) = -\nabla_x m^\dagger(x, y)\big(m - m^\dagger(x, y)\big). \tag{4.39}$$

The identity is derived by noting that, recalling (4.37) for the definition of $h^\dagger$, and then using (4.38),

$$\begin{aligned}
\frac{d}{dt}m &= h^\dagger(x, y)\\
&= \nabla_x m^\dagger(x, y)[f_0(x) + m^\dagger(x, y)] + \nabla_y m^\dagger(x, y) g^\dagger(x, y)\\
&= \nabla_x m^\dagger(x, y)[f_0(x) + m)] + \nabla_y m^\dagger(x, y) g^\dagger(x, y)\\
&\qquad - \nabla_x m^\dagger(x, y)\big(m - m^\dagger(x, y)\big)\\
&= \frac{d}{dt}m^\dagger(x, y) - \nabla_x m^\dagger(x, y)\big(m - m^\dagger(x, y)\big).
\end{aligned}$$

We emphasize this calculation is performed under the dynamics defined by (4.38).

The proof of the RNN approximation property proceeds by approximating vector fields $g^\dagger(x, y)$, $h^\dagger(x, y)$ by neural networks and introducing linear transformations of y and m to rewrite the approximate version of system (4.38) in the form (4.36). The effect of the approximation of the vector fields on the true solution is then propagated through the system and its effect controlled via a straightforward Gronwall argument.

Remark 4.5.16. The details of training continuous-time RNNs to ensure accuracy and long-time stability are a subject of current research [72, 125, 304, 82] and in this paper we confine the training of RNNs to an example in the general setting, and not the case of linear coupling. Discrete-time RNN training, on the other hand, is much more mature, and has produced satisfactory accuracy and stability for settings with uniform sample rates that are consistent across train and testing scenarios [165]. The form with linear coupling is widely studied in discrete time models. Furthermore, sophisticated variants on RNNs, such as Long-Short Term Memory (LSTM) RNNs [177] and Gated Recurrent Units (GRU) [87], are often more effective, although similar in nature RNNs. However, the potential formulation, implementation and advantages of these variants in the continuous-time setting [299] is not yet understood. We refer readers to [149] for background on discrete RNN implementations and backpropagation through time (BPTT). For implementations of continuous-time RNNs, it is common to leverage the success of the automatic BPTT code written in PyTorch and Tensorflow by discretizing (4.36) with an ODE solver that is compatible with these autodifferentiation tools (e.g. `torchdiffeq` by [358], `NbedDyn` by [304], and `AD-ENKF` by [82]). This compatibility can also be achieved by use of explicit Runge-Kutta schemes [332]. Note that the discretization of (4.36) can (and perhaps should) be much finer than the data sampling rate Δt, but that this requires reliable estimation of $x(t), \dot{x}(t)$ from discrete data. □

Remark 4.5.17. The need for data assimilation [21, 227, 346] to learn the initialization of recurrent neural networks may be understood as follows. Since $m^\dagger$ is not known and y is not observed (and in particular $y(0)$ is not known) the desired initialization for (4.38), and thus also for approximations of this equation in which $g^\dagger$ and $h^\dagger$ are replaced by neural networks, is not known. Hence, if an RNN is trained on a particular trajectory, the initial condition that is required for accurate approximation of (4.8) from an unseen initial condition is not known. Furthermore the invariant manifold $m = m^\dagger(x, y)$ may be unstable under numerical approximation. However if some observations of the trajectory starting at the new initial condition are used, then data assimilation techniques can potentially learn the initialization for the RNN and also stabilize the invariant manifold. Ad hoc initialization methods are common practice [170, 88, 26, 315], and rely on forcing the learned RNN with a short sequence of observed data to synchronize the hidden state. The success of these approaches likely rely on RNNs' abilities to emulate data assimilators [166]; however, a more careful treatment of the initialization problem may enable substantial advances. □

Remark 4.5.18. Reservoir computing (RC) is a variant on RNNs which has the advantage of leading to a quadratic optimization problem [190, 272, 154]. Within the context of the continuous-time RNN (4.36) they correspond to randomizing (A, B, c) in (4.36b) and then choosing only parameter C to fit the data. To be concrete, this leads to

$$r(t) = \mathcal{G}_t\big(\{x(s)\}_{s=0}^{t};\ r(0), A, B, c\big);$$

here $\mathcal{G}_t$ may be viewed as a random function of the path-history of x upto time t and of the initial condition for r. Then C is determined by minimizing the quadratic function

$$\mathcal{J}_T(C) = \frac{1}{2T}\int_0^T \|\dot{x}(t) - f_0(x(t)) - Cr(t)\|^2\, dt + \frac{\lambda}{2}\|C\|^2.$$

This may be viewed as a random feature approach on the Banach space X_T; the use of random features for learning of mappings between Banach spaces is studied by Nelsen and Stuart [295], and connections between random features and reservoir computing were introduced by Dong et al. [109]. In the specific setting described here, care will be needed in choosing probability measure on (A, B, c) to ensure a well-behaved map $\mathcal{G}_t$; furthermore data assimilation ideas [21, 227, 346] will be needed to learn an appropriate $r(0)$ in the prediction phase, as discussed in Remark 4.5.17 for RNNs. □

4.6 Numerical Experiments

In this section, we present numerical experiments intended to test different hypotheses about the utility of hybrid mechanistic and data-driven modeling. We summarize our findings in Section 4.6.1. We define the overarching experimental setup in Section 4.6.2.1, then introduce our criteria for evaluating model performance in Section 4.6.2.2. In the Lorenz '63 (L63) experiments (Section 4.6.2.3), we investigate how a simple Markovian random features model error term can be recovered using discrete and continuous-time methods, and how those methods scale with the magnitude of error, data sampling rate, availability of training data, and number of learned parameters. In the Lorenz '96 Multiscale (L96MS) experiments (Section 4.6.2.4), we take this a step further by learning a Markovian random features closure term for a scale-separated system, as well as systems with less scale-separation. As expected, we find that the Markovian closure approach is highly accurate for a scale-separated regime. We also see that the Markovian closure has merit even in cases with reduced scale-separation. However, this situation would clearly benefit from learning a closure term with memory, a topic we turn to in

Section 4.6.3, where we demonstrate that non-Markovian closure models can be learnt from noisy, partially observed data; for low-dimensional cases (e.g. L63), our method of training converges to return a good model with high short-term accuracy and long-term statistical validity. For higher-dimensional cases (e.g. L96MS), we find the method to hold promise, but further research is required in this general area. In Section 4.6.4, we demonstrate why non-Markovian closures must be carefully initialized and/or controlled (e.g. via data assimilation) in order to ensure their long-term stability and short-term accuracy.

4.6.1 Summary of Findings from Numerical Experiments

1. We find that hybrid modeling has better predictive performance than purely data-driven methods in a wide range of settings (see Figures 4.2a and 4.2b of Section 4.6.2.3): this includes scenarios where f_0 is highly accurate (but imperfect) and scenarios where f_0 is highly inaccurate (but nevertheless faithfully encodes much of the true structure for $f^\dagger$).

2. We find that hybrid modeling is more data-efficient than purely data-driven approaches (Figure 4.3 of Section 4.6.2.3).

3. We find that hybrid modeling is more parameter-efficient than purely data-driven approaches (Figure 4.4 of Section 4.6.2.3).

4. Purely data-driven discrete-time modeling can suffer from instabilities in the small timestep limit $\Delta t \ll 1$; hybrid discrete-time approaches can alleviate this issue when they are built from an integrator Ψ_0, as this will necessarily encode the correct parametric dependence on $\Delta t \ll 1$ (Figure 4.5 of Section 4.6.2.3).

5. In order to leverage standard supervised regression techniques, continuous-time methods require good estimates of derivatives $\dot{x}(t)$ from the data. Figure 4.5 of Section 4.6.2.3 quantifies this estimation as a function of data sample rate.

6. Non-Markovian model error can be captured by Markovian terms in scale-separated cases. Section 4.6.2.4 demonstrates this quantitatively in Figure 4.6, and qualitatively in Figure 4.7. Beyond the scale-separation limit, Markovian terms will fail for trajectory forecasting. However, Markovian terms may still reproduce invariant statistics in dissipative systems (for example, in cases with short memory-length). Section 4.6.2.4 demonstrates this quantitatively in Figure 4.6; Figure 4.7 offers intuition for these findings.

7. Non-Markovian description of model error is needed to accurately represent problems where the hidden dynamics is not scale-separated from the observed dynamics. Section 4.6.3 shows how partial and noisy observations can be exploited by augmented ODE models of form (4.18) when the noise and hidden dynamics are learnt implicitly by auto-differentiable data assimilation. We observe high-quality reconstruction of the L63 system along its first-component when choosing a correct (Figure 4.8) or overly enlarged (Figure 4.9) hidden dimension. We also observe promising reconstruction of the L96MS system in its slow components (Figure 4.10); however, long-time solutions to the learnt model exhibited instabilities inconsistent with the true system.

8. Non-Markovian models must be carefully initialized, and indeed data assimilation is needed, in order to ensure accuracy (Section 4.6.4) of invariant statistics (Figure 4.12), long-term stability (Figure 4.13), and accurate short-term predictions (Figure 4.14). We explain observed phenomena in terms of the properties of the desired lower-dimensional invariant manifold which is embedded within the higher dimensional system used as the RNN's basis of approximation.

4.6.2 Learning Markovian Model Errors from Noise-Free Data

4.6.2.1 Experimental Set-Up

In the Markovian error modeling experiments described in Sections 4.6.2.3 and 4.6.2.4, whether using continuous- or discrete-time models, we train a random features model on noise-free trajectories from the true system (an ODE). The problems we study provably have a compact global attractor and are provably (L63) or empirically (L96MS) ergodic; the invariant distribution is supported on the global attractor and captures the statistics of long-time trajectories which, by ergodicity, are independent of initial condition. The data trajectories are generated using scipy's implementation of Runge-Kutta 5(4) (via `solve_ivp`) with absolute and relative tolerances both 10^{-9} and maximum step size 10^{-4} [111, 420]. In order to obtain statistical results, we create 5 training trajectories from the true system of interest with initial conditions sampled independently from its attractor. Note that each training trajectory is long enough to explore the attractor, and each is used to train a separate model; the purpose is to observe the variance in learnt models with respect to randomly sampled paths through the attractor. We use the same sampling procedure to generate short independent validation and testing trajectories—we use 7 validation trajectories and 10 testing trajectories (these are short because we only use them to evaluate a model's short term forecast performance; when assessing long-term statistics of a learnt model, we compare to very long simulations from the true system). All plots use error bars to represent empirical estimates of the mean and standard deviation of the presented performance metric, as computed by ensembling the performance of the 5 models (one per training trajectory) over the 10 testing trajectories for a total of 70 random performance evaluations.

Each training procedure also involves an independent draw of the random feature functions as defined in (4.54). A validation step is subsequently performed to optimize the hyperparameters ω, β, as well as the regularization parameter λ. We automate this validation using Bayesian Optimization [291, 300], and find that it typically identifies good hyperparameters within 30 iterations. The entire process of entraining a model to a single, long training trajectory (including hyperparameter validation) typically takes approximately 30 minutes on a single core of a 2.1GHz Skylake CPU with an allocated 1GB RAM. Given a realization of random features and an optimal λ, we obtain the minimizer C^* using the Moore-Penrose Pseudoinverse implemented in scipy (`pinv2`). This learned C^*, paired with its random feature realization, is then used to predict 10 unseen testing trajectories (it is given the true initial condition for each of these testing trajectories).

When implementing in continuous-time, given high frequency but discrete-time data, two computational issues must be addressed: (i) extrapolation of the data to continuous-time; (ii) discretization of the resulting integrals. The approach we adopt avoids “inverse crimes” in which favourable behaviour is observed because of agreement between the data generation mechanism (with a specific integrator) and the approximation of the objective functions [93, 198, 438]; see Queiruga et al. [332] for further illustration of this issue and Keller and Du [204] and Du et al. [114] for a rigorous analysis of this inversion process in the context of linear multistep integration methods for deep learning. We interpolate the data with a spline, to obtain continuous-time trajectories, and then discretize the integrals using a simple Riemann sum; this strikes a desirable balance between robustness and efficiency and avoids inverse crimes. The discrete-time approaches, however, are able to learn not only model-discrepancy, but also integrator-based discrepancies; hence, the discrete-time methods may artificially appear to outperform continuous-time approaches, when, in fact, their performances might simply be considered to be comparable.

4.6.2.2 Evaluation Criteria

Models are evaluated against the test set for their ability to predict individual trajectories, as well as invariant statistics (the invariant measure and the autocorrelation function).

Trajectory Validity Time: Given threshold $\gamma > 0$, we find the first time t_γ at which the norm of discrepancy between true and approximate solutions reaches γ:

$$t_\gamma = \operatorname*{arg\,min}_{t \in [0,T]} \left\{ t : \ \|x(t) - x_m(t)\| \geq \gamma \overline{\|x(t)\|} \right\},$$

where $x(t)$ is the true solution to (4.2), $x_m(t)$ is the learned approximation, and the normed time average $\overline{\|x(t)\|}$ is approximated from training data. If the threshold is not violated on $[0, T]$, we define $t_\gamma := T$; this is rare in practice. We take $\gamma = 0.05$ (i.e. 5% relative divergence).

Invariant Distribution: To quantify errors in our reconstruction of the invariant measure, we consider the Kullback-Leibler (KL) divergence [220] between the true invariant measure μ and the invariant measure produced by our learned model μ_m. We approximate the divergence

$$d_{\mathrm{KL}}(\mu, \mu_m) := \int_{\mathbb{R}} \log \left(\frac{d\mu}{d\mu_m} \right) d\mu$$

by integrating kernel density estimates with respect to the Lebesgue measure.

Autocorrelation: We compare the autocorrelation function (ACF) with respect to the invariant distribution of the true and learned models. We approximate the ACF using a fast-fourier-transform for convolutions [378], and compare them via a normalized L_2 norm of their difference.

4.6.2.3 Lorenz '63 (L63)

Setting The L63 system [262] is described by the following ODE

$$\begin{aligned} \dot{u}_x &= a(u_y - u_x) \\ \dot{u}_y &= bu_x - u_y - u_x u_z \\ \dot{u}_z &= -cu_z + u_x u_y \end{aligned} \tag{4.40}$$

whose solutions are known to exhibit chaotic behavior for parameters $a = 10,\ b = 28,\ c = \frac{8}{3}$. We align these equations with our framework, starting from equation (4.1), by letting $x = (u_x, u_y, u_z)^T$ and defining $f^\dagger(x)$ to be the vector field appearing on the right-hand-side in (4.40). We define a discrete solution operator $\Psi^\dagger$ by numerical integration of $f^\dagger$ over a fixed time window Δt corresponding to a uniform data sampling rate, so that the true system is given by (4.1) in continuous-time and (4.6a) in discrete-time.

To simulate scenarios in which our available physics are good, but imperfect, we assume there exists additive unknown model error of form

$$m^\dagger(x) = \epsilon\, m_1(x) \tag{4.41}$$

with function m_1 determining the structure of model error, and scalar coefficient ϵ determining its magnitude. Recall that $f^\dagger = f_0 + m^\dagger$ and we assume f_0 is known to us. Our task is then to learn $f^\dagger$ by learning $m^\dagger$ and adding it to f_0. The discrete solution operator Ψ_0 is obtained as in (4.6b) by numerical integration of f_0 over a fixed time window Δt.

To simplify exposition, we explicitly define $m^\dagger$, then let $f_0 := f^\dagger - m^\dagger$. We first consider the setting where

$$m_1(x) := \begin{bmatrix} 0 \\ bu_x \\ 0 \end{bmatrix} \tag{4.42}$$

(as in [316]) and modulate ϵ in (4.41) to control the magnitude of the error term. In this case, f_0 can be viewed as the L63 equations with perturbed parameter $\tilde{b} = b(1-\epsilon)$, where b is artificially decreased by $100\epsilon\%$.

Then, we consider a more general case of heterogeneous, multi-dimensional residual error by drawing m_1 from a zero-mean Gaussian Process (GP) with a radial basis kernel (lengthscale 10). We form a map from $\mathbb{R}^3$ into itself by constructing three independent draws from a scalar-valued GP on $\mathbb{R}^3$. The resulting function is visualized in two-dimensional projections in Figure 4.1.

Observe that in the continuous-time framing, changes to ϵ do not impact the complexity of the learned error term; however, it does grow the magnitude of the error term. In the discrete-time framing, larger values of ϵ can magnify the complexity of the discrepancy $\Psi_0(x) - \Psi^\dagger(x)$.

Results We perform a series of experiments with the L63 system in order to illustrate key points about using data to learn model errors in dynamical systems. First, we demonstrate that hybrid modeling tends to outperform data-only and physics-only methods in terms of prediction quality. We first consider model error

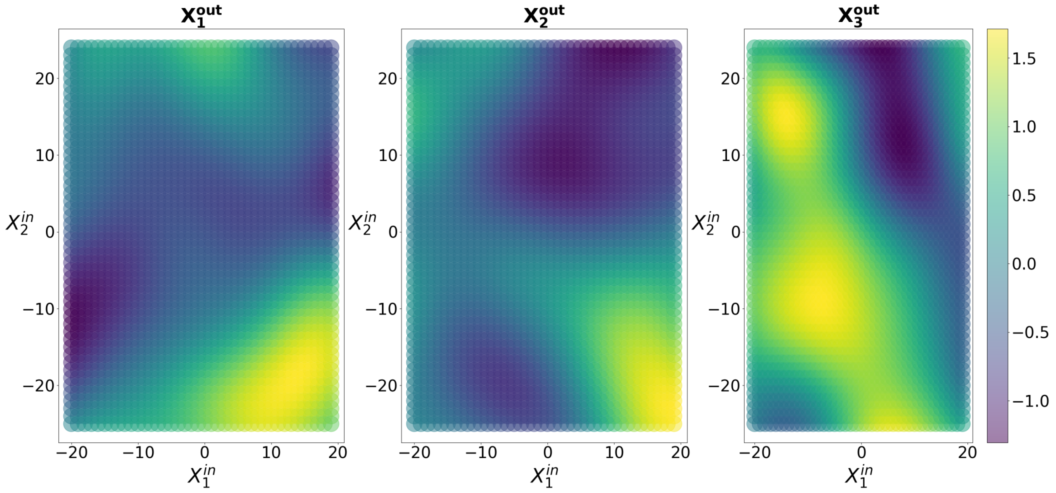


Figure 4.1: Here we visualize an example of the function m_1 in (4.41), which is obtained as a single random draw from a zero-mean Gaussian Process mapping $\mathbb{R}^3 \to \mathbb{R}^3$. We have plotted its output surface as three scalar functions (left to right) of the first two inputs (the plot axes) with the third input component fixed at 0.

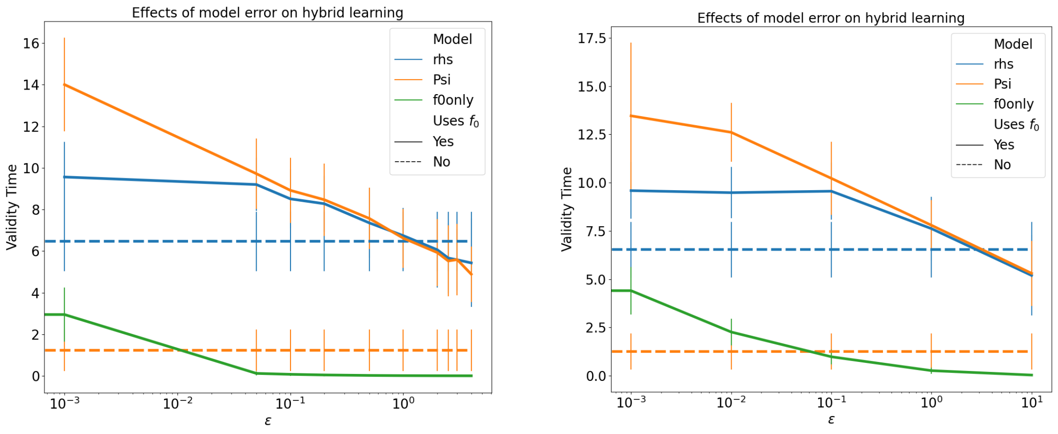
as in (4.42); see Figure 4.2a in which we study performance (validity time) versus model error amplitude (ϵ), using random feature maps with $D = 200$, and a single trajectory of length $T = 100$ sampled at timestep $\Delta t = 0.001$. Unless otherwise specified, this is also the configuration used in subsequent experiments.

We see identical trends in Figure 4.2b for a more general case with the non-parametric model error term constructed from Gaussian processes. Interestingly, we see that for small and moderate amounts of model error ϵ, the hybrid methods substantially outperform data-only and physics-only methods. Eventually, for large enough model discrepancy, the hybrid-methods and data-only methods have similar performance; indeed the hybrid-method may be outperformed by the data-only method at large discrepancies. For the simple model error this appears to occur when the discrepancy term is larger in magnitude than f_0 (e.g. for $b = 28$ and $\epsilon = 2$, the model error term $\epsilon b u_x$ can take on values larger than $f^\dagger$ itself).

Figure 4.2b also shows that a continuous-time approach is favored over discrete-time when using data-only methods, but suggests the converse in the hybrid modeling context. We suspect this is an artifact of the different integration schemes used in data generation, training, and testing phases; the data are generated with a higher-fidelity integrator than the one available in training and testing. For the continuous-time method, this presents a fundamental limitation to the forecast quality (we chose this to avoid having artificially high forecast validity times). However, the discrete-time method can overcome this by not only learning the mechanistic model discrepancy, but also the discrepancy term associated with a mis-matched integrator. This typically happens when a closure is perfectly learnable and deterministic (i.e. our Lorenz '63 example); in this case, the combination of physics-based and integrator-sourced closures can be learned nearly perfectly. In later experiments with a multiscale system, the closures are considered approximate (they model the mean of a noisy underlying process) and the discrete- and continuous-time methods perform more similarly, because the inevitable imperfections of the learned closure term dominate the error rather than the mis-specified integrator. Note that approximate closures driven by scale-separation are much more realistic; thus we should not expect the hybrid discrete-time method to dramatically outperform hybrid continuous-time methods unless other limitations are present (e.g. slow sampling rate).

Importantly, the parameter regime for which hybrid methods sustain advantage over the imperfect physics-only method is substantial; the latter has trajectory predictive performance which drops off rapidly for very small ϵ. This suggests that an apparently uninformative model can be efficiently modified, by machine learning techniques, to obtain a useful model that outperforms a *de novo* learning approach.

Next, we show that hybrid methods simplify the machine learning task in terms of complexity of the learned function and, consequently, the amount of data needed for the learning. Figure 4.3 examines prediction performance (validity time) as a function of training data quantity using random feature maps with $D = 2000$ and a fixed parametric model error ($\epsilon = 0.2$ in (4.41)) and sampling rate $\Delta t = 0.01$. We see that the hybrid methods substantially outperform the data-only approaches in regimes with limited training data. For the continuous-time example, we see an expected trend, where the data-only methods are able to catch up to the hybrid methods with the acquisition of more data. The discrete-time models do not exhibit this behavior, but we expect the data-only discrete-time model to eventually catch up, albeit with additional training data and number of parameters. Note that greater expressivity is also required from data-only methods—our choice of a large $D = 2000$ aims to give all methods ample expressivity, and thus test convergence with respect to training data quantity alone. These results demonstrate that the advantage of hybrid modeling is magnified when training data are limited and cannot fully inform *de novo* learning. Figure 4.4 further studies the impact of expressivity by again fixing a parametric model error ($\epsilon = 0.05$ in (4.41)), training length $T = 100$, and sampling rate $\Delta t = 0.001$. We see that all methods improve with a larger number of



(a) (b)

Figure 4.2: These plots shows the temporal length of the forecast validity of our learnt models of L63 (4.40), each as a function of model error, as parameterized by ϵ (4.41) (with $D = 200$, $T = 100$, and $\Delta t = 0.001$). Continuous-time methods are shown in blue, discrete-time approaches in orange. Dotted lines indicate purely data-driven methods to learn the entire vector field defining the dynamics; solid lines indicate methods that learn perturbations to the imperfect mechanistic models f_0 or Ψ_0. Integration using the imperfect mechanistic model, without recourse to data, is shown in green. In Figure 4.2a, we employ the linear form of model error m_1 defined in (4.42). In Figure 4.2b, we let m_1 be a single draw from a Gaussian Process, whose structure is shown in Figure 4.1. Here, we plot means, with error bars as 1 standard deviation.

random features, but that relative superiority of hybrid methods is maintained even for $D = 10000$.

Finally, we study trade-offs between learning in discrete- versus continuous-time for the L63 example (4.40). Figure 4.5 examines prediction performance (validity time) as a function of data sampling rate Δt using random feature maps with $D = 200$ with a fixed parametric model error ($\epsilon = 0.05$ in (4.41)) and an abundance of training data $T = 1000$. We observe that for fast sampling rates ($\Delta t < 0.01$), the continuous-time and discrete-time hybrid methods have similar performance. For $\Delta t > 0.01$, derivatives become difficult to estimate from the data and the performance of the continuous-time methods rapidly decline. However, the discrete-time methods sustain their predictive performance for slower sampling rates ($\Delta t \in (0.01, 0.1)$). At some point, the discrete-time methods deteriorate as well, as the discrete map becomes complex to learn at longer terms because of the sensitivity to initial conditions that is a hallmark of chaotic systems. Here, the discrete-time methods begin to fail around $\Delta t = 0.2$; note that they can be extended to longer time intervals by increasing D and

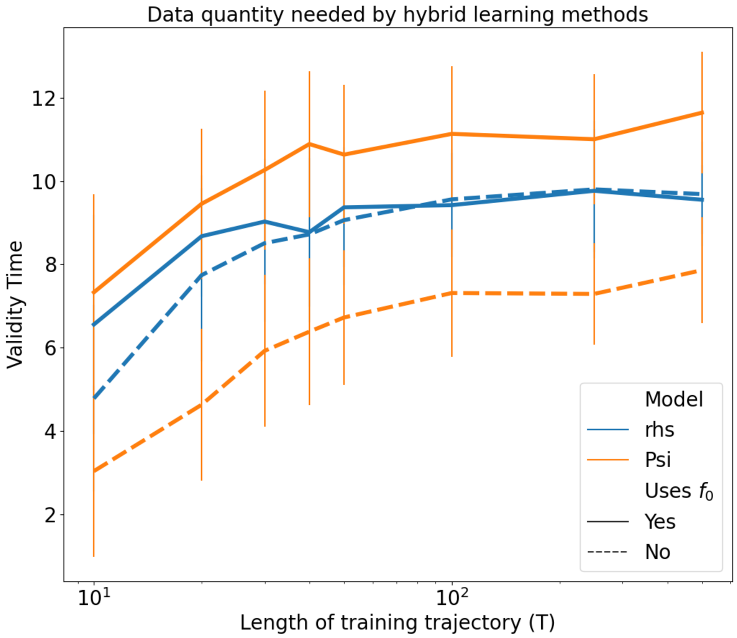


Figure 4.3: Here we examine the performance of the proposed methods as a function of the length of the interval over which the training data is provided, where $\Delta t = 0.01$, ($\epsilon = 0.2$ in (4.41)), and $D = 2000$ are held constant for the L63 example (4.40). See description of Figure 4.2 for explanation of legend. We observe that all methods improve with increasing training lengths. We see that, in continuous-time, the primary benefit in hybrid modeling is when the training data are limited.

amount of training data, but returns diminish quickly.

4.6.2.4 Lorenz '96 Multiscale (L96MS) System

Setting Here, we consider the multiscale system [260] of form (4.7), where each variable $X_k \in \mathbb{R}$ is coupled to a subgroup of fast variables $Y_k \in \mathbb{R}^J$. We have $X \in \mathbb{R}^K$ and $Y \in \mathbb{R}^{K \times J}$. For $k = 1 \dots K$ and $j = 1 \dots J$, we write

$$\dot{X}_k = f_k(X) + h_x \overline{Y}_k \tag{4.43a}$$

$$\dot{Y}_{k,j} = \frac{1}{\varepsilon} r_j(X_k, Y_k) \tag{4.43b}$$

$$\overline{Y}_k = \frac{1}{J} \sum_{j=1}^{J} Y_{k,j} \tag{4.43c}$$

$$f_k(X) = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F \tag{4.43d}$$

$$r_j(X_k, Y_k) = -Y_{k,j+1}(Y_{k,j+2} - Y_{k,j-1}) - Y_{k,j} + h_y X_k \tag{4.43e}$$

$$X_{k+K} = X_k, \quad Y_{k+K,j} = Y_{k,j}, \quad Y_{k,j+J} = Y_{k+1,j} \tag{4.43f}$$

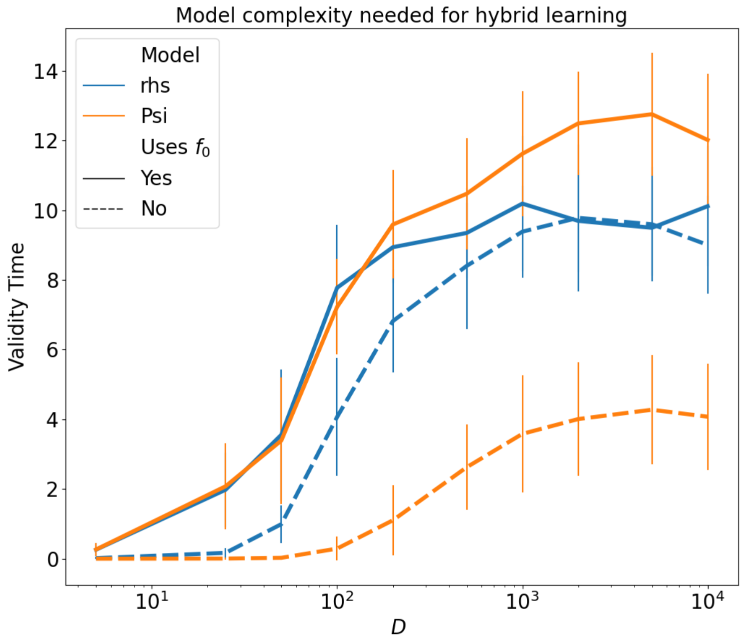


Figure 4.4: Here we examine the performance of the proposed methods as a function of model complexity, where $\Delta t = 0.001$, $\epsilon = 0.05$, and $T = 100$ are held constant for the L63 example (4.40). See description of Figure 4.2 for explanation of legend. We observe that all methods improve with increasing number of parameters, and that hybrid methods are especially beneficial when available complexity is limited.

where $\varepsilon > 0$ is a scale-separation parameter, $h_x, h_y \in \mathbb{R}$ govern the couplings between the fast and slow systems, and $F > 0$ provides a constant forcing. We set $K = 9,\ J = 8,\ h_x = -0.8,\ h_y = 1,\ F = 10$; this leads to chaotic dynamics for ε small. When studying scale-separation, we consider $\varepsilon \in \{2^{-7},\ 2^{-5},\ 2^{-3},\ 2^{-1}\}$.

We consider the setting in which we learn Markovian random features models in variable X alone, from X data generated by the coupled (X, Y) system. Large scale-separation between the observed (X) and unobserved (Y) spaces can simplify the problem of accounting for the unobserved components; in particular, for sufficient scale-separation, we expect a Markovian term to recover a large majority of the residual errors. In fact, we further simplify this problem by learning a scalar-valued model error M that is applied to each X_k identically in the slow system:

$$\dot{X}_k \approx f_k(X) + M(X_k).$$

This choice stems from observations about statistical interchangeability amongst the slow variables of the system; these properties of the L96MS model in the scale-separated regime are discussed in [132]. We can directly align our reduction of

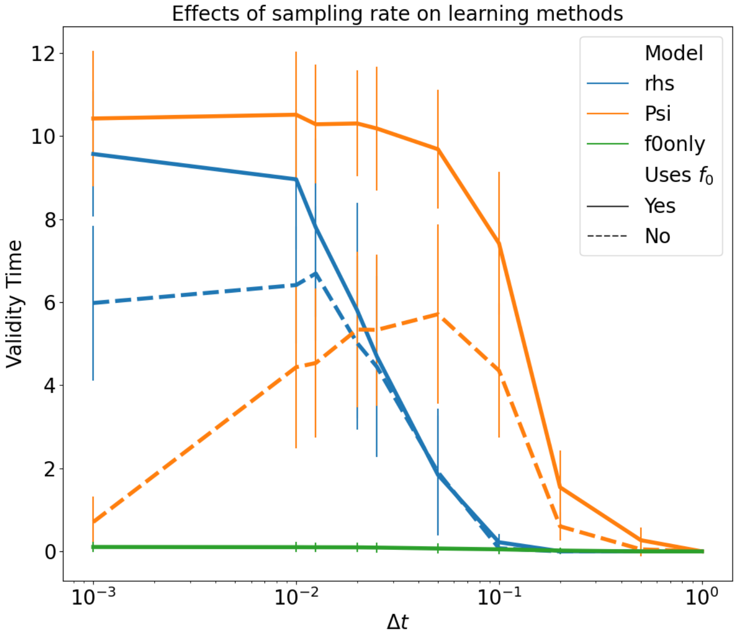


Figure 4.5: This shows temporal forecast validity as a function of the step size of training data for the tested methods in the L63 example (4.40). We hold fixed $D = 200$, $\epsilon = 0.05$, and $T = 1000$. See description of Figure 4.2 for explanation of legend. We see that while purely data-driven discrete-time methods struggle at short time steps, the hybrid version thrives in this scenario. All approaches, of course, eventually decay as large time steps create more complex forward maps, due to sensitivity to initial conditions. We also see continuous-time methods work well for small time steps, then deteriorate in tandem with quality of estimated derivatives.

(4.43) with the Markovian hybrid learning framework in (4.2) as follows:

$$\begin{aligned}
\dot{X} &\approx f_0(X) + m(X) \\
f_0(X) &:= [f_1(X), \cdots, f_K(X)]^T \\
m(X) &:= [M(X_1), \cdots, M(X_K)]^T.
\end{aligned}$$

Results We plot the performance gains of our hybrid learning approaches in Figure 4.6 by considering validity times of trajectory forecasts, estimation of the invariant measure, and ACF estimation. In all three metrics (and for all scale-separations ε), *de novo* learning in discrete ($\Psi^\dagger \approx m$) and continuous-time ($f^\dagger \approx m$) is inferior to using the nominal mechanistic model f_0. We found that the amount of data used in these experiments is insufficient to learn the full system from scratch. On the other hand, hybrid models in discrete ($\Psi^\dagger \approx \Psi_0 + m$) and continuous-time ($f^\dagger \approx f_0 + m$) noticeably outperformed the nominal physics.

Surprisingly, Figure 4.6 shows that the Markovian closure methods still qualitatively reproduce the invariant statistics even for large ε settings where we would expect substantial memory effects. Figure 4.6 also demonstrates this quantitatively using KL-divergence between invariant measures and mean-squared-error between ACFs. It seems that for this dissipative system, memory effects tend to average out in the invariant statistics. However, the improvements in validity time for trajectory-based forecasting deteriorate for $\varepsilon = 2^{-1}$.

To visualize this non-Markovian structure, and how it might be exploited, we examine the residuals from f_0 in Figure 4.7 and observe that there are discernible trajectories walking around the Markovian closure term. For small ε, these trajectories oscillate rapidly around the closure term. For large ε (e.g. 2^{-1}), however, we observe a slow-moving residual trajectory around the Markovian closure term. This indicates the presence of a stronger memory component, and thus would benefit from a non-Markovian approach to closure modeling.

Jiang and Harlim [193] show that the memory component in this setting with $\varepsilon = 2^{-1}$ can be described using a closure term with a simple delay embedding of the previous state at lag 0.01. They learn the closure using a kernel method cast in an RKHS framework, for which random feature methods provide an approximation.

4.6.3 Learning From Partial, Noisy Observations

In this section, we focus on the non-Markovian setting outlined in Section 4.2.3, and attempt to model the dynamics of the observable using (4.18), with f_1, f_2 given by two-layer, fully connected neural networks with GeLU activations [172], and perform the learning by minimizing (4.22) from Section 4.4.3.3, using 3DVAR for the data assimilation [227, 224], with the ADAM optimizer [209]. The learning rate was initialized at 0.01 and tuned automatically using a scheduler that halved the learning rate if the training error had not decreased over 10 (mini-batched) epochs. Data were sampled at $\Delta t = 0.01$ in all cases, and normalized to have mean zero and unit variance. Numerical integration was performed with the `torchdiffeq` implementation of the Dormand-Prince adaptive fifth-order Runge-Kutta method: for the L63 example, simple backpropagated autodifferentiation was performed through this solver; for the L96MS example, we used the adjoint method provided by [358].

4.6.3.1 Lorenz '63

We first consider modeling the dynamics of the first-component of the L63 system in (4.40), where we noisily observe the first-component – that is, we observe a

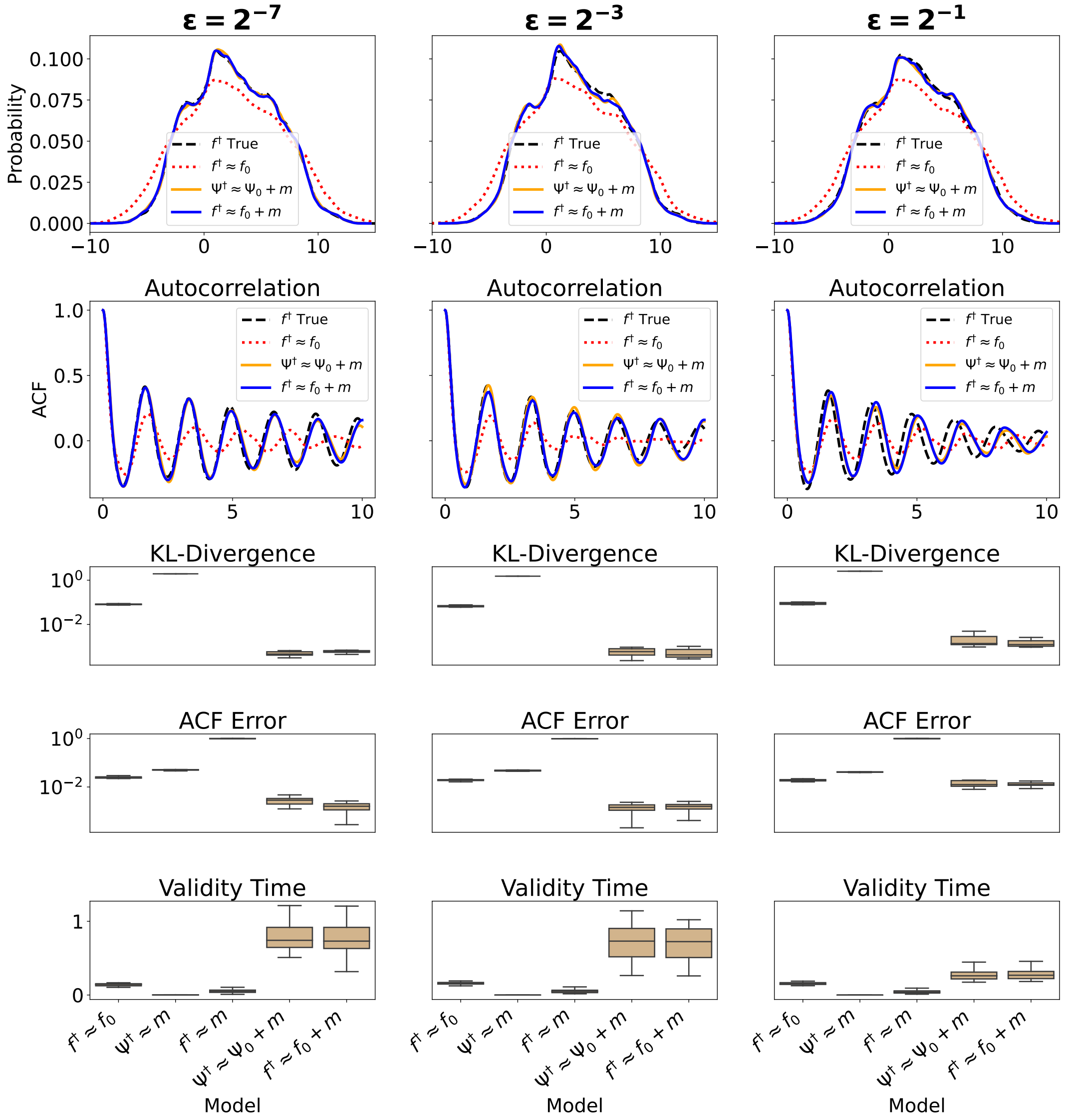


Figure 4.6: This figure shows the performance of different approaches to modeling the L96MS slow subsystem (4.43). In $f^\dagger \approx f_0$, we only use the nominal physics f_0. In $\Psi^\dagger \approx m$ and $f^\dagger \approx m$, we try to learn the entire right-hand-side using only data (in discrete- and continuous-time settings, respectively). In $\Psi^\dagger \approx \Psi_0 + m$ and $f^\dagger \approx f_0 + m$, we focus on learning Markovian residuals for the known physics (in discrete- and continuous-time settings, respectively). The residual-based correctors substantially outperform the nominal physics and purely data-driven methods according to all presented metrics: invariant measure (shown qualitatively in the first row and quantitatively in the third row), ACF (shown qualitatively in the second row and quantitatively in the fourth row), and trajectory forecasts (shown in the final row). The boxplots show the distributions of quantitative metrics (e.g. KL-divergence, squared errors, validity time), which come from different models, each trained on a different trajectory, and generated using an independent random feature set. Notably, the Markovian residual-based methods' performance deteriorates for small scale-separation ($\varepsilon = 2^{-1}$), where the Markovian assumption breaks down.

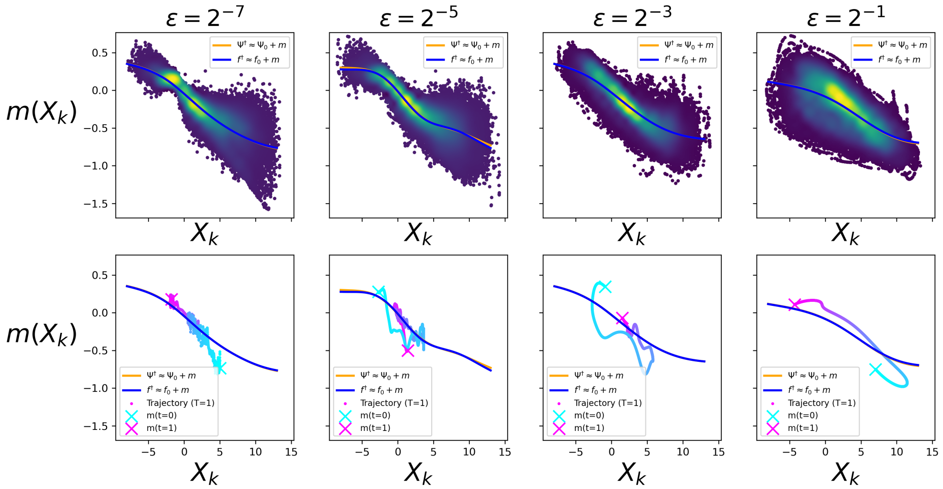


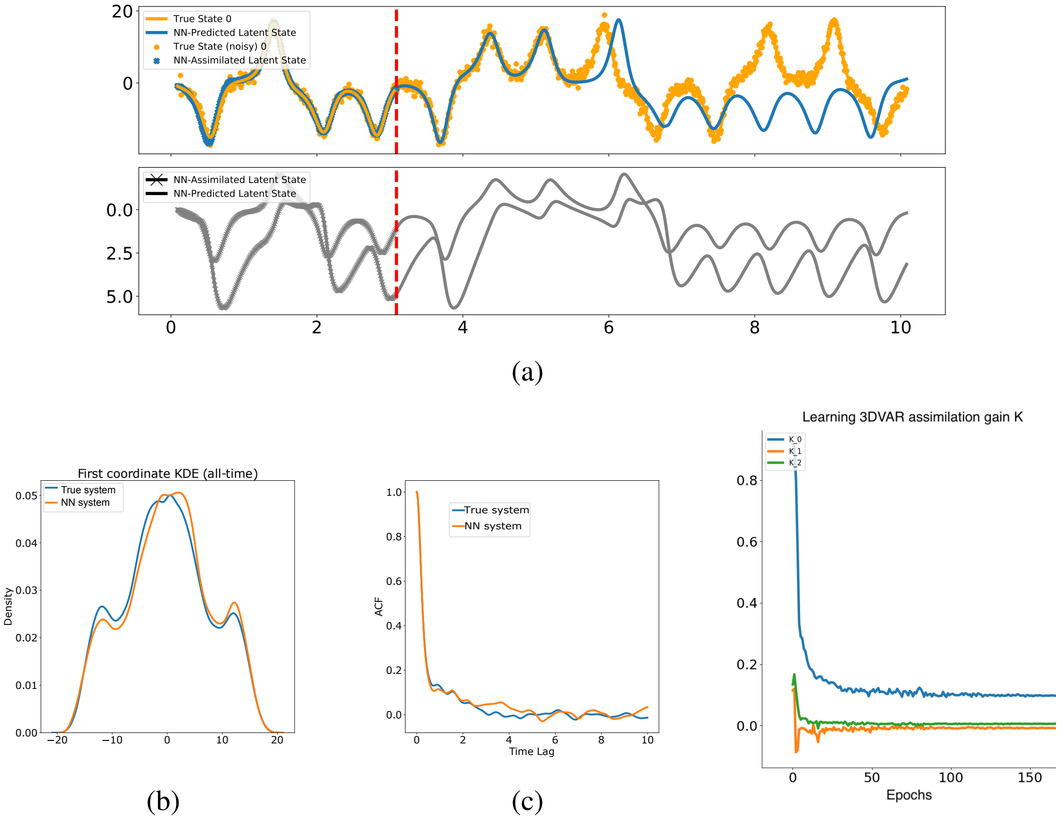
Figure 4.7: This figure shows the observed and estimated residuals of the nominal physics model f_0 for the L96MS slow subsystem (4.43) at different scale-separation factors. The first row shows the density of these residuals (yellow is high density, blue is low), as well as the fit of our closure terms in continuous- (blue) and discrete- (orange) time (the discrete model was normalized by dividing by Δt). The second row shows temporal structure in the errors of our residual fit by superimposing a short (T=1) one-dimensional trajectory (this represents $\sim 0.1\%$ of training data).

noisy trajectory of u_x (i.i.d. additive zero-mean, variance-one Gaussian), but do not observe the remaining components u_y, u_z. We jointly trained on 100 trajectories, each of length $T = 10$ and randomly initialized from a box around the attractor; we chose this approach to ensure that we had data coverage both on and off the attractor although we note that similar success is obtained with a single trajectory of length $T = 1000$.). The neural network had width 50. We chose an assimilation time of $\tau_1 = 3$ and a forecast time of $\tau_2 = 0.1$. The optimization ran for approximately 200 epochs, and took roughly 24hrs on a single GPU. Adequate results were obtained using a fixed 3DVAR gain matrix $K = [0.5, 0, 0]^T$. However, we present results using the algorithm in which $K = \theta_{\text{DA}}$ is jointly learned along with parameters θ_{DYN}, as described in Section 4.4.3.3; this demonstrates that the gain need not be known *a priori*.

First, we present results using knowledge that the correct hidden dimension $d_r = 2$: in Figure 4.8a, we show an example of the trained model being assimilated (using 3DVAR with learnt K) during the first 3 time units, then predicting for another 7 time units; recall that training was performed using only a $\tau_2 = 0.1$ forecasting horizon, but we evaluate on a longer time horizon to provide a robust test metric. Observe that the learnt hidden dynamics in gray are synchronized with the data, then used to perform the forecast. In Figures 4.8b and 4.8c, we show that by solving the system for long time (here, $T = 10^4$), we are able to accurately reproduce invariant statistics (invariant measure and autocorrelation, resp.) for the true system. In Figure 4.8d, we show the evolution of the learnt K.

Next, we let $d_r = 10$, exceeding the true dimension of the hidden states; thus we are able to explore issues caused by learning an overly expressive (in terms of dimension of hidden states) dynamical model. Figure 4.9 shows dynamics for a learnt model in this setting; we found its reproduction of invariant statistics to be similar to the cases in Figures 4.8b and 4.8c, but omit the plots for brevity. This success aligns with the approximation theory, as discussed in Remark 4.5.11, and provides empirical reassurance that the methodology can behave well in situations where the dimension of the hidden variable is unknown and dimension d_r used in learning exceeds its true dimension. Nevertheless, we construct an example in Section 4.6.4 in which a specific embedding of the true dynamics in a system of higher dimension can lead to poor approximation; this is caused by an instability in the model which allows departure from the invariant manifold on which the true dynamics is accurately captured. However, we emphasize that this phenomenon is not observed empirically in the experiment reported here with $d_r = 10$. Nonetheless we also note expected decreases in efficiency caused by over-estimating the dimension of the hidden variable, during both model training and testing; thus determining the smallest choice for d_r, compatible with good approximation, is important. Recent research has addressed this challenge in the discrete-time setting by applying manifold learning to a delay-embedding space, then using the learnt manifold to inform initialization and dimensionality of LSTM hidden states [205].

Note that our early attempts at achieving these numerical results, using the optimization ideas in Sections 4.4.3.1 and 4.4.3.2, yielded unstable models that exhibited blow-up on shorter time scales (e.g. $T < 1000$); however, by incorporating data assimilation as in [82], and further tuning the optimization to achieve lower training errors, we were able to obtain a model that, empirically, did not exhibit blow-up, even when solved for very long time (e.g. $T = 10^5$). We also note that we were unable to achieve such high-fidelity results using the methods of [304] on neural networks with non-linear activation functions; this may be explained by noting that Ouala et al. [304] achieved their results using linear, identity-based activations, resulting in inference of polynomial models containing the true L63 model.



(a)

(b)

(c)

(d)

Figure 4.8: This figure concerns learning of a continuous-time RNN to model the L63 system (4.40), based on noisy observation of only the first component; it uses an augmented state space $d_r = 2$. Figure 4.8a shows how the trained model can be used for forecasting—by first synchronizing to data using 3DVAR, then forecasting into the future. The top-half depicts dynamics of the observed component (model-solutions in blue; observations in yellow); the bottom-half depicts the augmented state space (both hidden components are shown in gray). We observed a validity time of roughly 3 model time units. Figures 4.8b and 4.8c shows that long-time solutions of the learnt model accurately mirror invariant statistics (invariant measure and autocorrelation, resp.) for the true system. Figure 4.8d shows the learning process for estimating a 3DVAR gain K.

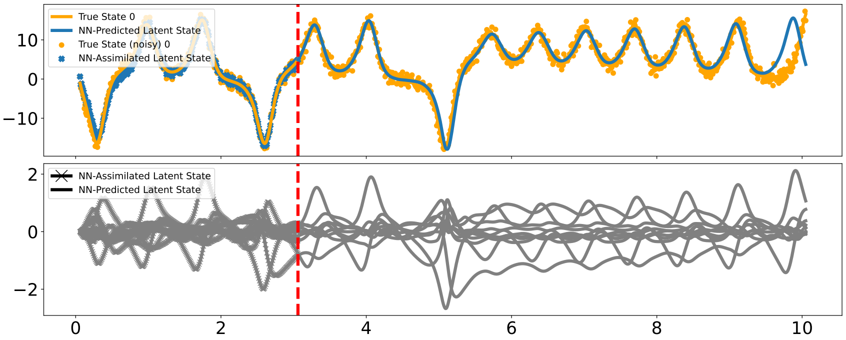
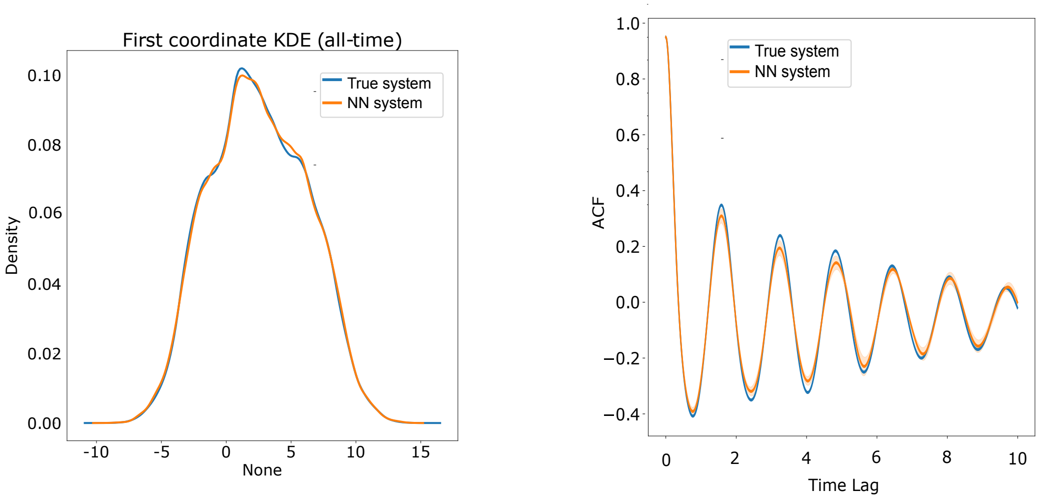


Figure 4.9: This figure concerns learning of a continuous-time RNN to model the L63 system (4.40), based on noisy observation of only the first component; it uses an augmented state space $d_r = 10$. The top-half depicts dynamics of the observed component (model-solutions in blue; observations in yellow); the bottom-half depicts the augmented state space (all 10 hidden components are shown in gray). In the first 3 time units, the model is assimilated to a sequence of observed data using 3DVAR, then in the subsequent 7 time units, a forecast is issued. We found this model to have similar short-term and long-term fidelity when compared to the model presented in Figures 4.8a to 4.8d, which used the correct hidden dimension $d_r = 2$.

4.6.3.2 Lorenz '96 Multiscale ($\varepsilon = 2^{-1}$)

Recall that Markovian closures fail to capture autocorrelation statistics for the slow components of this model in the case of $\varepsilon = 2^{-1}$ (see top right panel of Figure 4.6). As evidenced by the slow-moving trajectory around the Markovian closure in Figure 4.7, this is a case ripe for non-Markovian modeling. We investigate the applicability of our continuous-time ODE formulation in (4.18), using a neural network of width of 1000. We applied the above described methodology for minimizing (4.22), under the data setting described in Section 4.6.2.4, to learn hidden dynamics. Similarly to the previous section, we jointly trained on 100 trajectories, each of length $T = 20$ and randomly initialized from a box around the attractor. We chose an assimilation time of $\tau_1 = 2$ and a forecast time of $\tau_2 = 1$; note that longer times can become quite costly, especially for high-dimensional systems; nevertheless, the assimilation time τ_1 appears intrinsically tied to the amount of memory present in the system.

In Figures 4.10a and 4.10b, we plot comparisons of the true and learnt (via (4.18)) ACF and invariant measure, and observe substantial improvement over the Markovian closure. However, this learnt model exhibited instabilities when solved for longer than $T = 500$. We expect that this can be remedied via further training (as was found for the L63 example); however, the incorporation of stability constraints into the model, as in [375], would be valuable. In order to train this larger model for longer



(a)

(b)

Figure 4.10: This figure concerns learning of a continuous-time RNN to model the first 9 (slow) components of the L96MS system ($\varepsilon = 0.5$ in (4.43)), based on noisy observations of these slow components; it uses an augmented state space $d_r = 72$. We trained using noised observations (standard deviation 0.01) of only the first 9 components of the true 81−dimensional system. These plots show that this model can accurately reproduce both the invariant measure (Figure 4.10a) and ACF (Figure 4.10b) for these observed states. These statistics were calculated by running the learnt model for $T = 500$ model time units; longer runs encountered instabilities that caused trajectories to leave the attractor and blow-up.

time, further studies of efficient optimization must also be performed in this setting ([82] has begun highly relevant investigation in this direction).

In Figure 4.11, we visualize the learnt 3DVAR gain (which encodes the learnt model's covariance structure), in which each row corresponds to the gain for a given component of the learnt model as a function of observed components (indexed in the columns); trends are elucidated via hierarchical clustering and a row-based normalization of the learnt matrix K. It clearly learns a consistent diagonal covariance structure for the observables. More impressively, it illustrates cross-covariances between observed and hidden components that mirror the compartmentalized structure of the model in (4.43); note that each observed component has a distinct grouping of hidden variables which have high correlation (white) primarily with that component and low correlation (black) with other observables. This type of analysis may provide greater interpretability of learnt models of hidden dynamics.

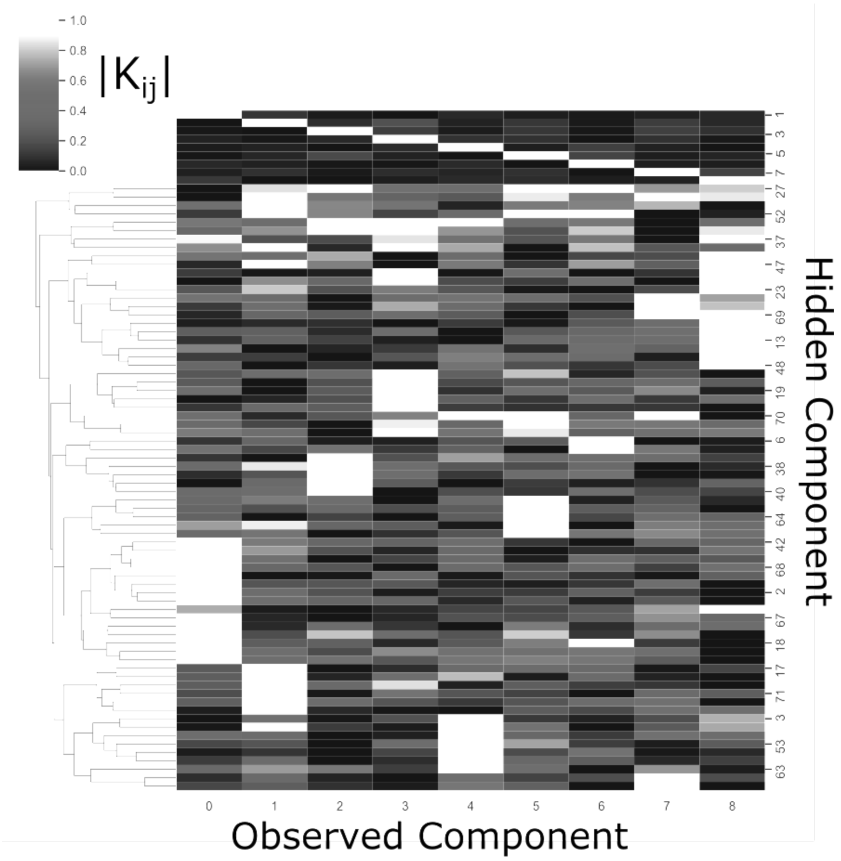


Figure 4.11: Here we visualize the learnt 3DVAR gain matrix K (81 x 9) (θ_{DA} in Section 4.4.3.3) associated with the non-Markovian learning of L96MS (4.43). We first compute entry-wise absolute values, then apply a row-normalization; white indicates highest correlation, and black indicates lowest correlation. The top 9 rows shown directly correspond to the first 9 rows of K. The bottom 72 rows are re-ordered (via hierarchical clustering) to illustrate associations between the 9 observed components and the 72 hidden variables.

4.6.4 Initializing and Stabilizing the RNN

As mentioned in Remark 4.5.17 the RNN approximates an enlarged system which contains solutions of the original system as trajectories confined to the invariant manifold $m = m^\dagger(x, y)$; see identity (4.39). However, this invariant manifold may be unstable, either as a manifold within the continuous-time model (4.38), or as a result of numerical instability. We now demonstrate this with numerical experiments. This instability points to the need for data assimilation to be used with RNNs if prediction of the original system is desired, not only to initialize the system but also to stabilize the dynamics to remain near to the desired invariant manifold.

To illustrate these challenges, we consider the problem of modeling evolution of a single component of the L63 system (4.40). Consider this as variable x in (4.7). As exhibited in (4.38), model error may be addressed in this setting by learning a representation that contains the hidden states y in (4.7) (i.e. the other two unobserved components of (4.40)), but since the dimension of the hidden states is typically not known *a priori* the dimensions of the latent variables in the RNN (and the system it approximates) may be greater than those of y; in the specific construction we use to prove the existence of an approximating RNN we introduce a vector field for evolution of the error m as well as y. We now discuss the implications of embedding the true dynamics in a higher dimensional system in the specific context of the embedded system (4.38). However the observations apply to any embedding of the desired dynamics (4.7) (with $\epsilon = 1$) within any higher dimensional system.

We choose examples for which (4.39) implies that $m - m^\dagger$ is constant in time. Then, under (4.38),

$$\big(m - m^\dagger(x, y)\big)(t) = \text{constant};$$

that is, it is constant in time. The desired invariant manifold (where the constant is 0) is thus stable. However this stability only holds in a neutral sense: linearization about the manifold exhibits a zero eigenvalue related to translation of $m - m^\dagger$ by a constant. We now illustrate that this embedded invariant manifold can be unstable; in this case the instability is caused by numerical integration, which breaks the conservation of $m - m^\dagger$ in time.

Example 1: Consider equation (4.40) which we write in form (4.8) by setting $x = u_x$ and $y = (u_y, u_z)$. Then we let $f_0(u_x) := -au_x$ yielding $m^\dagger(u_y) = au_y$. Thus $f^\dagger = f_0 + m^\dagger$ is defined by the first component of the right-hand side of (4.40). The function $g^\dagger(u_y, u_z)$ is then given by the second and third components of the right-hand side of (4.40). Applying the methodology leading to (4.38) to (4.40) results in the following four dimensional system:

$$\dot{u}_x = f_0(u_x) + m, \qquad u_x(0) = x_0, \tag{4.44a}$$

$$\dot{u}_y = bu_x - u_y - u_x u_z, \qquad u_y(0) = y_0, \tag{4.44b}$$

$$\dot{u}_z = -cu_z + u_x u_y, \qquad u_z(0) = z_0, \tag{4.44c}$$

$$\dot{m} = a\big(bu_x - u_y - u_x u_z\big), \qquad m(0) = m^\dagger(y_0). \tag{4.44d}$$

Here we have omitted the u_y-dependence from the equation (4.40) for u_x, and aim to learn this error term; we introduce the variable m in order to do so. This system, when projected into u_x, u_y, u_z, behaves identically to (4.40) when $m(0) = m^\dagger(y_0)$.

Thus the 4−dimensional system in (4.44) has an embedded invariant manifold on which the dynamics is coincident with that of the 3−dimensional L63 system.

We numerically integrate the 4−dimensional system in (4.44) for 10000 model time units (initialized at $x_0 = 1, y_0 = 3, z_0 = 1, m_0 = ay_0 = 30$), and show in Figure 4.12 that the resulting measure for u_x (dashed red) is nearly identical to its invariant measure in the traditional 3−dimensional L63 system in (4.40) (solid black). However, we re-run the simulation for a perturbed $m(0) = m^\dagger(y_0) + 1$, and see in Figure 4.12 (dotted blue) that this yields a different invariant measure for u_x. This result emphasizes the importance of correctly initializing an RNN not only for efficient trajectory forecasting, but also for accurate statistical representation of long-time behavior.

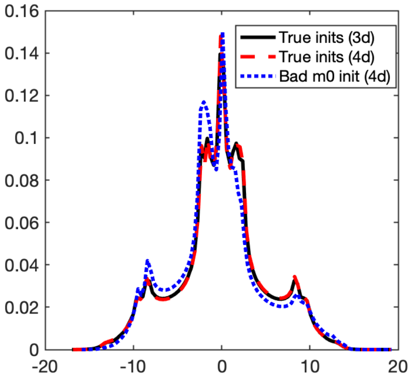


Figure 4.12: Here, we show that the invariant density for the first component of L63 (black) can be reproduced by a correctly initialized augmented 4−d system (dashed red) in (4.44). However, incorrect initialization of $m(0)$ in (4.44) (dotted blue) yields a different invariant density.

Example 2: Now we consider (4.40) which we write in form (4.8) by setting $x = u_z$ and $y = (u_x, u_y)$. We let $f_0(u_z) := -cu_z$ and $m^\dagger(u_x, u_y) := u_x u_y$, so that $f^\dagger = f_0 + m^\dagger$ corresponds to the third component of the right-hand side of (4.40). Function $g^\dagger(u_x, u_y)$ is defined by the first two components of the right-hand side of (4.40). We again form a 4−dimensional system corresponding to (4.40) using the methodology that leads to (4.38):

$$\dot{u}_x = a(u_y - u_x), \qquad u_x(0) = x_0, \tag{4.45a}$$

$$\dot{u}_y = bu_x - u_y - u_x u_z, \qquad u_y(0) = y_0, \tag{4.45b}$$

$$\dot{u}_z = f_0(u_z) + m, \qquad u_z(0) = z_0, \tag{4.45c}$$

$$\dot{m} = u_x \dot{u}_y + u_y \dot{u}_x, \qquad m(0) = m^\dagger(x_0, y_0). \tag{4.45d}$$

We integrate (4.45) for 3000 model time units (initialized at $x_0 = 1, y_0 = 3, z_0 = 1, m_0 = x_0 y_0 = 3$), and show in Figure 4.13 that the 3−dimensional Lorenz attractor is unstable with respect to perturbations in the numerical integration of the 4−dimensional system. The solutions for u_x, u_y, u_z eventually collapse to a fixed point after the growing discrepancy between $m(t)$ and $m^\dagger$ becomes too large. The time at which collapse occurs may be delayed by using smaller tolerances within the numerical integrator (we employ `Matlab rk45`) demonstrating that the instability is caused by the numerical integrator. This collapse is very undesirable if prediction of long-time statistics is a desirable goal. On the other hand, Figure 4.14 shows short-term accuracy of the 4−dimensional system in (4.45) up to 12 model time units when correctly initialized ($m_0 = m^\dagger(x_0, y_0)$, dashed red), and accuracy up to 8 model time units when initialization of m_0 is perturbed ($m_0 = m^\dagger(x_0, y_0) + 1$, dotted blue). This result demonstrates the fundamental challenges of representing chaotic attractors in enlarged dimensions and may help explain observations of RNNs yielding good short-term accuracy, but inaccurate long-term statistical behavior. While empirical stability has been observed in some discrete-time LSTMs [421, 165], the general problem illustrated above is likely to manifest in any problems where the dimension of the learned model exceeds that of the true model; the issue of how to address initialization of such models, and its interaction with data assimilation, therefore merits further study.

4.7 Conclusions

In this work we evaluate the utility of blending mechanistic models of dynamical systems with data-driven methods, demonstrating the power of hybrid approaches. We provide a mathematical framework that is consistent across parametric and non-parametric models, encompasses both continuous- and discrete-time, and allows for Markovian and memory-dependent model error. We also provide basic theoretical results that underpin the adopted approaches. The unified framework elucidates commonalities between seemingly disparate approaches across various applied and theoretical disciplines. It would be desirable if the growing recognition of the need for hybrid modeling were to motivate flexible incorporation of mechanistic models into open-source software for continuous-time Markovian and non-Markovian modeling of error [304, 358, 72, 125, 17, 157].

Our work is focused on immutable mechanistic models (f_0 and Ψ_0), but these models themselves often have tunable parameters. In principle one can jointly learn parameters for the mechanistic model and closure term. However, the lack of

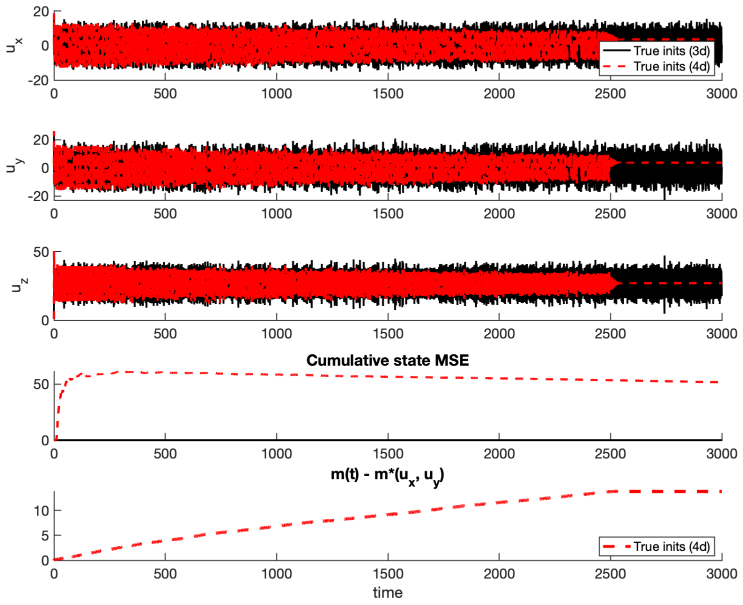


Figure 4.13: Here we show that the embedded 3−dimensional manifold of L63, within the 4−dimensional system given by (4.45), is unstable. Indeed the correctly initialized 4−dimensional system (dashed red) has solution which decays to a fixed point. The bottom figure shows divergence of the numerically integrated model error term $m(t)$ and the state-dependent term $m^\dagger$; this growing discrepancy is likely responsible for the eventual collapse of the 4−dimensional system.

identifiability between modifying the closure and modifying the physics brings up an interesting question in explainability. Future work might focus on decoupling the learning of parameters and closure terms so that maximal expressivity is first squeezed out of the mechanistic model [324, 323].

Our numerical results demonstrate the superiority of hybrid modeling over learning an entire system from scratch, even when the available mechanistic model has large infidelities. Hybrid modeling also showed surprisingly large performance gains over using mechanistic models with only small infidelities. We quantify these improvements in terms of data hunger, demands for model complexity, and overall predictive performance, and find that all three are significantly improved by hybrid methods in our experiments.

We establish bounds on the excess risk and generalization error that decay as $1/\sqrt{T}$ when learning model discrepancy from a trajectory of length T in an ergodic continuous-time Markovian setting. We make minimal assumptions about the nominal physics (i.e. $f_0 \in C^1$); thus, our result equivalently holds for learning the entire vector field $f^\dagger$ (i.e $f_0 \equiv 0$). However the upper bounds on excess risk and

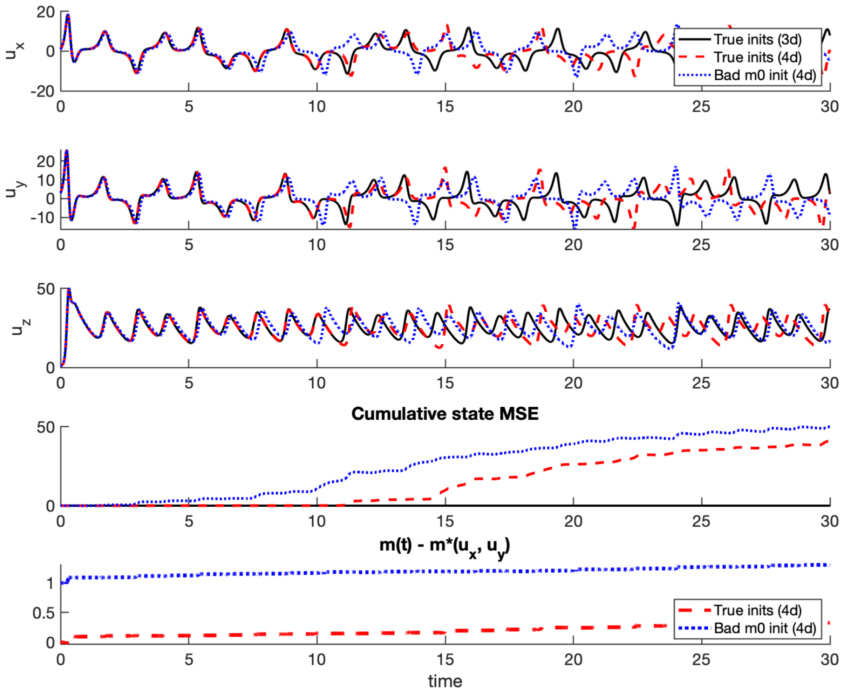


Figure 4.14: Here, we show short-term accuracy for the 4−dimensional system in (4.45). Predictions using the correct initialization of m_0 (dashed red) remain accurate for nearly twice as long as predictions that use a perturbed initialization ($m_0 = m^\dagger(u_x, u_y) + 1$). The bottom figure shows that $m(t)$ diverges from the state-dependent $m^\dagger$ more quickly for the poorly initialized model, but in both cases errors accumulate over time.

generalization error scale with the size of the function being learned, hence going some way towards explaining the superiority of hybrid modeling observed in the numerical experiments. Future theoretical work aimed at quantifying the benefits of hybrid learning versus purely data-driven learning is of interest. We also note that the ergodic assumption underlying our theory will not be satisfied by many dynamical models, and alternate statistical learning theories need to be developed in such settings.

We illustrate trade-offs between discrete-time and continuous-time modeling approaches by studying their performance as a function of training data sample rate. We find that hybrid discrete-time approaches can alleviate instabilities seen in purely data-driven discrete-time models at small timesteps; this is likely due to structure in the integrator Ψ_0, which has the correct parametric dependence on timestep. In the continuous-time setting, we find that performance is best when derivatives can accurately be reconstructed from the data, and deteriorates in tandem with differentiation inaccuracies (caused by large timesteps); continuous-time hybrid methods appear to offer additional robustness to inaccurate differentiation when compared to purely data-driven methods. In cases of large timesteps and poorly resolved derivatives, ensemble-based data assimilation methods may still allow for accurate learning of residuals to the flow field for continuous-time modeling [151].

Finally, we study non-Markovian memory-dependent model error, through numerical experiments and theory, using RNNs. We prove universal approximation for continuous-time hybrid RNNs and demonstrate successful deployment of the methodology. Future work focusing on the effective training of these models, for more complex problems, would be of great value; ideas from data assimilation are likely to play a central role [82]. Further work on theoretical properties of reservoir computing (RC) variants on RNNs would also be of value: they benefit from convex optimization, and may be viewed as random feature methods between Banach spaces. These RNN and RC methods will benefit from constraining the learning to ensure stability of the latent dynamical model. These issues are illustrated via numerical experiments that relate RNNs to the question of stability of invariant manifolds representing embedded desired dynamics within a higher dimensional system.

4.8 Supplementary Information

4.8.1 Proof of Excess Risk/Generalization Error Theorem

Note that in both (4.32) and (4.33) $\varphi(\cdot)$ is only evaluated on (compact) $\mathcal{A}$ obviating the need for any boundedness assumptions on the functions $\{f_\ell\}_{\ell=0}^p$ and $m^\dagger$ in what follows.

Lemma 4.8.1. *Let Assumptions 4.5.1 and 4.5.3 hold. Then there is Σ positive semi-definite symmetric in $\mathbb{R}^{p\times p}$ such that $\theta_T^* \to \theta_\infty^*$ almost surely, and $\sqrt{T}(\theta_T^* - \theta_\infty^*) \Rightarrow N(0,\Sigma)$ with respect to $x(0) \sim \mu$. Furthermore, there is constant $C \in (0,\infty)$ such that, almost surely w.r.t. $x(0) \sim \mu$,*

$$limsup_{T\to\infty}\left(\frac{T}{\log\log T}\right)^{\frac{1}{2}}\|\theta_T^* - \theta_\infty^*\| \le C.$$

Proof. By rearranging the equation for θ_∞^* we see that

$$\begin{aligned} A_T\theta_T^* &= b_T, \\ A_T\theta_\infty^* &= b_\infty + (A_T - A_\infty)\theta_\infty^*. \end{aligned}$$

Thus, subtracting,

$$(\theta_T^* - \theta_\infty^*) = A_T^{-1}(b_T - b_\infty) - A_T^{-1}(A_T - A_\infty)\theta_\infty^*. \tag{4.46}$$

Because $\{f_\ell(\cdot)\}$ and $m^\dagger(\cdot)$ are Hölder (Assumption 4.5.3, and discussion immediately preceding it), so are $\langle f_i(\cdot), f_j(\cdot)\rangle$ and $\langle m^\dagger(\cdot), f_j(\cdot)\rangle$. Thus each entry of matrix A_T (resp. vector b_T) converges almost surely to its corresponding entry in A_∞ (resp. b_∞), by the ergodicity implied by Assumption 4.5.1, and the pointwise ergodic theorem. The almost sure convergence of θ^*_T to θ^*_∞ follows, after noting that A_∞ is invertible. Furthermore, also by Assumption 4.5.1, there are constants $\{\sigma_{ij}\}, \{\sigma_j\}$ such that

$$\sqrt{T}\Big((A_T)_{ij} - (A_\infty)_{ij}\Big) \Rightarrow N(0, \sigma_{ij}^2),$$
$$\sqrt{T}\Big((b_T)_j - (b_\infty)_j\Big) \Rightarrow N(0, \sigma_j^2).$$

Since arbitrary linear combinations of the $\{(A_T)_{ij}\}, \{(b_T)_j\}$ are time-averages of Hölder functions, it follows that $\sqrt{T}\{A_T - A_\infty, b_T - b_\infty\}$ converges in distribution to a Gaussian, by the Cramér-Wold Theorem [155]. Weak convergence of $\sqrt{T}(\theta^*_T - \theta^*_\infty)$ to a Gaussian follows from (4.46) by use of the Slutsky Lemma [155], since A_T converges almost surely to invertible A_∞. Matrix Σ cannot be identified explicitly in terms of only the $\{\sigma_{ij}\}, \{\sigma_i\}$ because of correlations between A_T and b_T. The almost sure bound on $\|\theta^*_T - \theta^*_\infty\|$ follows from (4.46) after multiplying by $(T/\log\log T)^{\frac{1}{2}}$, noting that $A_T \to A_\infty$ almost surely, and the almost sure bounds on $(T/\log\log T)^{\frac{1}{2}}\{\|A_T - A_\infty\|, \|b_T - b_\infty\|\}$, using Assumption 4.5.1. □

In what follows it is helpful to define

$$R^+_T = (\theta^*_T - \theta^*_\infty)\big(\|\theta^*_T\| + \|\theta^*_\infty\| + 1\big),$$
$$G^+_T = \mathcal{I}_T(m^*_\infty) - \mathcal{I}_\infty(m^*_\infty).$$

Lemma 4.8.2. *Let Assumption 4.5.3 hold. Then, assuming $x(0) \sim \mu$, there is constant $C > 0$ such that the excess risk R_T satisfies*

$$R_T \le C\|R^+_T\|.$$

Furthermore the generalization error satisfies

$$|G_T| \le 2C\|R^+_T\| + |G^+_T|.$$

Proof. For the bound on the excess risk we note that

$$\begin{aligned} R_T &= \mathcal{L}_\mu(m^*_T, m^\dagger) - \mathcal{L}_\mu(m^*_\infty, m^\dagger) \\ &= \int_{\mathbb{R}^{d_x}} \Big\langle (m^*_T - m^*_\infty)(x), (m^*_T + m^*_\infty - 2m^\dagger)(x) \Big\rangle\, \mu(dx) \end{aligned}$$

$$\leq \Big(\int_{\mathbb{R}^{d_x}} \big\|(m_T^* - m_\infty^*)(x)\big\|^2 \mu(dx)\Big)^{\frac{1}{2}} \Big(\int_{\mathbb{R}^{d_x}} \big\|(m_T^* + m_\infty^* - 2m^\dagger)(x)\big\|^2 \mu(dx)\Big)^{\frac{1}{2}}.$$

The first follows from the boundedness of the $\{f_\ell\}_{\ell=1}^p$ and $m^\dagger$, since the first term in the product above is bounded by a constant multiple of $\|\theta_T^* - \theta_\infty^*\|$ and the second term by a constant multiple of $\|\theta_T^*\| + \|\theta_\infty^*\| + \sup_{\mathcal{A}} \|m^\dagger\|$.

For the bound on the generalization error we note that

$$\begin{aligned} G_T &= \mathcal{I}_T(m_T^*) - \mathcal{I}_\infty(m_T^*) \\ &= \mathcal{I}_T(m_T^*) - \mathcal{I}_T(m_\infty^*) \\ &\qquad\qquad + \mathcal{I}_T(m_\infty^*) - \mathcal{I}_\infty(m_\infty^*) \\ &\qquad\qquad\qquad\qquad + \mathcal{I}_\infty(m_\infty^*) - \mathcal{I}_\infty(m_T^*) \\ &= \big(\mathcal{I}_T(m_T^*) - \mathcal{I}_T(m_\infty^*)\big) + G_T^+ - R_T. \end{aligned}$$

The third term in the final identity is the excess risk that we have just bounded; the first term may be bounded in the same manner that we bounded the excess risk, noting that integration with respect to μ is simply replaced by integration with respect to the empirical measure generated by the trajectory data which, by assumption, is confined to the attractor $\mathcal{A}$; the second term is simply G_T^+. Thus the result follows. □

Proof of Theorem 4.5.4. By Assumption 4.5.1, with choice of $\varphi(x) = \|m^\dagger(x) - m_\infty^*(x)\|^2$, $\sqrt{T}G_T^+$ converges in distribution to a scalar-valued centred Gaussian. By Lemma 4.8.1 and the Slutsky Lemma [155], $\sqrt{T}R_T^+$ converges in distribution to a centred Gaussian in $\mathbb{R}^p$. By the Cramer-Wold Theorem [155] $\sqrt{T}(R_T^+, G_T^+)$ converges in distribution to a centred Gaussian in $\mathbb{R}^{p+1}$.

The convergence in distribution results for excess risk R_T and generalization error $|G_T|$ then follow from Lemma 4.8.2, under Assumption 4.5.1. Furthermore, by Lemma 4.8.1, there is constant $C_1 > 0$ such that

$$\operatorname{limsup}_{T\to\infty}\Big(\frac{T}{\log\log T}\Big)^{\frac{1}{2}} \|R_T^+\| \leq C_1;$$

similarly, possibly by enlarging C_1, Assumption 4.5.1 gives

$$\operatorname{limsup}_{T\to\infty}\Big(\frac{T}{\log\log T}\Big)^{\frac{1}{2}} |G_T^+| \leq C_1.$$

The desired almost sure bound on $R_T + |G_T|$ follows from Lemma 4.8.2.

□

4.8.2 Proof of Continuous-Time ODE Approximation Theorem (General Case)

Proof. Recall equations (4.38). By 4.5.9, for any $\delta_o > 0$ there exist dimensions N_g and N_m and parameterizations $\theta_g \in \mathbb{R}^{N_g}, \theta_m \in \mathbb{R}^{N_m}$ such that for any $\big(x, y\big) \in B(0, 2\rho_T)$, and in the maximum norm,

$$\|g^\dagger(x, y) - f_2(x, y;\ \theta_g)\| \leq \delta_o$$
$$\|m^\dagger(x, y) - f_1(x, y;\ \theta_m)\| \leq \delta_o.$$

By using these, we can rewrite (4.38) as

$$\begin{aligned}\dot{x} &= f_0(x) + f_1(x, y;\ \theta_m) + e_x(t)\\ \dot{y} &= f_2(x, y;\ \theta_g) + e_y(t)\end{aligned} \tag{4.47}$$

where, uniformly for $\big(x(0), y(0)\big) \in B(0, \rho_0)$,

$$\sup_{t\in[0,T]} \|e_y(t)\| \leq \delta_o$$
$$\sup_{t\in[0,T]} \|e_x(t)\| \leq \delta_o.$$

By removing the bounded error terms, we obtain the approximate system:

$$\begin{aligned}\dot{x}_\delta &= f_0(x_\delta) + f_1(x_\delta, y_\delta;\ \theta_m)\\ \dot{y}_\delta &= f_2(x_\delta, y_\delta;\ \theta_g)\end{aligned} \tag{4.48}$$

Next, we obtain a stability bound on the discrepancy between the approximate system (4.48) and the true system (originally written as (4.8) and re-formulated as (4.47)). First, let $w = (x, y)$, $w_\delta = (x_\delta, y_\delta)$ and define F to be the concatenated right-hand-side of (4.48). Note that F is $L-$Lipschitz in the maximum norm on $B(0, 2\rho_T)$, for some L related to the Lipschitz continuity of f_0, f_1, and f_2. Then we can write the true and approximate systems, respectively, as (using the maximum norm)

$$\dot{w} = F(w) + e_w(t) \tag{4.49a}$$
$$\dot{w}_\delta = F(w_\delta), \tag{4.49b}$$

where

$$\sup_{t\in[0,T]} \|e_w(t)\| \leq \sup_{t\in[0,T]} \|e_y(t)\| + \sup_{t\in[0,T]} \|e_x(t)\| \leq 2\delta_o.$$

Let $Pw = (x, y)$. Then, for any $t \in [0, T]$, and for all $Pw(0), Pw_\delta(0) \in B(0, \rho_0)$

$$\|w(t) - w_\delta(t)\| \leq \left\|w(0) - w_\delta(0)\right\| + \int_0^t \left\|e_w(s)\right\| ds + \int_0^t \left\|F\big(w(s)\big) - F\big(w_\delta(s)\big)\right\| ds.$$

This follows by writing (4.49) in integrated form, subtracting and taking norms. Using the facts that $\left\|e_w(s)\right\| \leq 2\delta_o$ and F is $L-$Lipschitz we obtain, for $t \in [0, T]$,

$$\left\|w(t) - w_\delta(t)\right\| \leq \left\|w(0) - w_\delta(0)\right\| + 2\delta_o T + L \int_0^t \left\|w(s) - w_\delta(s)\right\| ds.$$

By the integral form of the Gronwall Lemma, it follows that for all $t \in [0, T]$:

$$\left\|w(t) - w_\delta(t)\right\| \leq \left[\|w(0) - w_\delta(0)\| + 2\delta_o T\right] \exp(Lt).$$

Thus,

$$\sup_{t \in [0,T]} \left\|w(t) - w_\delta(t)\right\| \leq \left[\|w(0) - w_\delta(0)\| + 2\delta_o T\right] \exp(LT).$$

By choice of initial conditions and δ_o sufficiently small we can achieve a $\delta > 0$ approximation. Finally, we note that the approximate system (4.48) is a function of parameter $\theta_\delta = [\theta_m, \theta_g] \in \mathbb{R}^{N_\delta}$ with $n_\delta = N_g + N_m$. □

4.8.3 Proof of Continuous-Time RNN Approximation Theorem (Linear in Observation)

Proof. Recall equations (4.38). By approximation theory by means of two-layer feed-forward neural networks [95], for any $\delta_o > 0$ there exist embedding dimensions N_g and N_h and parameterizations

$$\begin{aligned}
\theta_g &= \{C_g \in \mathbb{R}^{d_y \times N_g},\ B_g \in \mathbb{R}^{N_g \times d_x},\ A_g \in \mathbb{R}^{N_g \times d_y},\ c_g \in \mathbb{R}^{N_g}\},\\
\theta_h &= \{C_h \in \mathbb{R}^{d_x \times N_h},\ B_h \in \mathbb{R}^{N_h \times d_x},\ A_h \in \mathbb{R}^{N_h \times d_y},\ c_h \in \mathbb{R}^{N_h}\}
\end{aligned}$$

such that for any $\big(x, y\big) \in B(0, 2\rho_T)$, and in the maximum norm,

$$\begin{aligned}
\|g^\dagger(x, y) - C_g \sigma(B_g x + A_g y + c_g)\| &\leq \delta_o\\
\|h^\dagger(x, y) - C_h \sigma(B_h x + A_h y + c_h)\| &\leq \delta_o.
\end{aligned}$$

Without loss of generality we may assume that C_g and C_h have full rank since, if they do not, arbitrarily small changes can be made which restore full rank. By using these parameterizations and embedding dimensions, we can rewrite (4.38) as

$$\begin{aligned}
\dot{x} &= f_0(x) + m\\
\dot{y} &= C_g \sigma(B_g x + A_g y + c_g) + e_y(t)\\
\dot{m} &= C_h \sigma(B_h x + A_h y + c_h) + e_{m^\dagger}(t)
\end{aligned} \tag{4.50}$$

where, uniformly for $\big(x(0), y(0)\big) \in B(0, \rho_0)$,

$$\sup_{t\in[0,T]} \|e_y(t)\| \leq \delta_o$$
$$\sup_{t\in[0,T]} \|e_{m^\dagger}(t)\| \leq \delta_o.$$

By removing the bounded error terms, we obtain the approximate system:

$$\begin{aligned} \dot{x}_\delta &= f_0(x_\delta) + m_\delta \\ \dot{y}_\delta &= C_g\sigma(B_g x_\delta + A_g y_\delta + c_g) \\ \dot{m}_\delta &= C_h\sigma(B_h x_\delta + A_h y_\delta + c_h) \end{aligned} \tag{4.51}$$

Here $m_\delta(t)$ is initialized at $m^\dagger(x(0), y(0))$. Next, we obtain a stability bound on the discrepancy between the approximate system (4.51) and the true system (originally written as (4.8) and re-formulated as (4.50)). First, let $w = (x, y, m)$, $w_\delta = (x_\delta, y_\delta, m_\delta)$ and define F to be the concatenated right-hand-side of (4.51). Note that F is $L-$Lipschitz in the maximum norm, for some L related to the Lipschitz continuity of f_0, approximation parameterization θ_δ, and regularity of nonlinear activation function σ. Then we can write the true and approximate systems, respectively, as

$$\dot{w} = F(w) + e_w(t) \tag{4.52a}$$
$$\dot{w}_\delta = F(w_\delta), \tag{4.52b}$$

where

$$\sup_{t\in[0,T]} \|e_w(t)\| \leq \sup_{t\in[0,T]} \|e_y(t)\| + \sup_{t\in[0,T]} \|e_{m^\dagger}(t)\| \leq 2\delta_o.$$

Let $Pw = (x, y)$ and $P^\perp w = m$; recall that $P^\perp w(0)$ is defined in terms of $Pw(0)$. Then, for any $t \in [0, T]$, and for all $Pw(0), Pw_\delta(0) \in B(0, \rho_0)$

$$\|w(t) - w_\delta(t)\| \leq \big\|w(0) - w_\delta(0)\big\| + \int_0^t \big\|e_w(s)\big\|ds + \int_0^t \big\|F\big(w(s)\big) - F\big(w_\delta(s)\big)\big\|ds.$$

By following the logic in Section 4.8.2, we have

$$\sup_{t\in[0,T]} \big\|w(t) - w_\delta(t)\big\| \leq \big[\|w(0) - w_\delta(0)\| + 2\delta_o T\big] \exp(LT).$$

By choice of initial conditions and δ_o sufficiently small we can achieve a $\delta > 0$ approximation.

Finally, we note that the approximate system (4.51) may be written as a recurrent neural network of form (4.36) as follows. Consider the equations

$$\begin{aligned} \dot{x}_\delta &= f_0(x_\delta) + C_h n_\delta \\ \dot{z}_\delta &= \sigma(B_g x_\delta + A_g C_g z_\delta + c_g) \\ \dot{n}_\delta &= \sigma(B_h x_\delta + A_h C_g z_\delta + c_h) \end{aligned} \tag{4.53}$$

where we have defined (z_δ, n_δ) in terms of (y_δ, m_δ) by $y_\delta = C_g z_\delta$ and $m_\delta = C_h n_\delta$. Now note that (4.53) is equivalent to (4.36), with recurrent state r_δ and parameters θ_δ given by:

- $r_\delta = \begin{bmatrix} z_\delta \\ n_\delta \end{bmatrix}$
- $C_\delta = \begin{bmatrix} 0 & C_h \end{bmatrix}$
- $B_\delta = \begin{bmatrix} B_g \\ B_h \end{bmatrix}$
- $A_\delta = \begin{bmatrix} A_g C_g & 0 \\ A_h C_g & 0 \end{bmatrix}$
- $c_\delta = \begin{bmatrix} c_g \\ c_h \end{bmatrix}$

Any initial condition on $(y_\delta(0), m_\delta(0))$ may be achieved by choice of initializations for $(z_\delta(0), n_\delta(0))$, since C_g, C_h are of full rank.

□

4.8.4 Random Feature Approximation

Random feature methods lead to function approximation for mappings between Hilbert spaces $X \to Y$. They operate by constructing a probability space $(\Theta, \nu, \mathcal{F})$ with $\Theta \subseteq \mathbb{R}^p$ and feature map $\varphi\colon X \times \Theta \to Y$ such that $k(x, x') := \mathbb{E}^{\vartheta}[\varphi(x;\ \vartheta) \otimes \varphi(x';\ \vartheta)] \in \mathcal{L}(Y, Y)$ forms a reproducing kernel in an associated reproducing kernel Hilbert space (RKHS) K. Solutions are sought within $\operatorname{span}\{\varphi(\cdot\ ;\ \vartheta_l)\}_{l=1}^m$ where the $\{\vartheta_l\}$ are picked i.i.d. at random. Theory supporting the approach was established in finite dimensions by Rahimi and Recht [335]; the method was recently applied in the infinite dimensional setting in [295].

We now explain the precise random features setting adopted in Section 4.5, and hypothesis classes given by (4.26) and (4.31). We start with random feature functions $\varphi(\cdot\,;\,\vartheta)\colon \mathbb{R}^{d_x} \to \mathbb{R}$, with $\vartheta = [w, b]$,

$$\begin{aligned} w \in \mathbb{R}^{d_x} &\sim \mathcal{U}(-\omega, \omega) \\ b \in \mathbb{R} &\sim \mathcal{U}(-\beta, \beta) \\ \varphi(x;\, w, b) &:= \tanh(w^T x + b), \end{aligned} \tag{4.54}$$

and $\omega, \beta > 0$. We choose D i.i.d. draws of w, b, and stack the resulting random feature functions to form the map $\phi(x)\colon \mathbb{R}^{d_x} \to \mathbb{R}^D$ given by

$$\phi(x) := \Big[\varphi(x;\, w_1, b_w) \quad \dots \quad \varphi(x;\, w_D, b_D)\Big]^T.$$

We define hypothesis class (4.26) by introducing matrix $C\colon \mathbb{R}^D \to \mathbb{R}^{d_x}$ and seeking approximation to model error in the form $m(x) = C\phi(x)$ by optimizing a least squares function over matrix C. This does not quite correspond to the random features model with $X = Y = \mathbb{R}^{d_x}$ because, when written as a linear span of vector fields mapping $\mathbb{R}^{d_x}$ into itself, the vector fields are not independent. Nonetheless we found this approach convenient in practice and employ it in our numerics.

To align with the random features model with $X = Y = \mathbb{R}^{d_x}$, we choose $D = d_x$ and draw p functions $\phi(\cdot)$, labelled as $\{f_\ell(\cdot)\}$ i.i.d. at random from the preceding construction, leading to hypothesis class (4.31): we then seek approximation to model error in the form $m(x) = \sum_{\ell=1}^{p} \theta_\ell f_\ell(x)$. We find this form of random features model most convenient to explain the learning theory perspective on model error.

4.8.5 Derivation of Tikhonov-Regularized Linear Inverse Problem

Here, we show that optimization of (4.27)

$$\mathcal{J}_T(C) = \frac{1}{2T}\int_0^T \big\|\dot{x}(t) - f_0(x) - C\phi\big(x(t)\big)\big\|^2\, dt + \frac{\lambda}{2}\|C\|^2$$

reduces to a Tikhonov-regularized linear inverse problem. Since (4.30) is quadratic in C, there exists a unique global minimizer for C^* such that $\frac{\partial \mathcal{J}_T}{\partial C}(C^*) = 0$. The minimizer C^* satisfies:

$$(Z + \lambda I)C^T = Y$$

where

$$\begin{aligned} Z &= \overline{[\phi \otimes \phi]_T} \\ Y &= \overline{[\phi \otimes (\dot{x} - f_0)]_T}. \end{aligned}$$

and

$$\begin{aligned} [A \otimes B]_t &:= A(t)B^T(t) \\ \overline{A_T} &:= \frac{1}{T}\int_0^T A(t)dt \end{aligned}$$

for $A(t) \in \mathbb{R}^{m\times n},\ B(t) \in \mathbb{R}^{m\times l}$.

To see this, observe that

$$\begin{aligned} \mathcal{J}_T(C) &= \frac{1}{2T}\int_0^T \|\dot{x}(t) - f_0\big(x(t)\big) - C\phi(x(t))\|^2 dt + \frac{\lambda}{2}\|C\|^2 \\ &= \frac{1}{2T}\int_0^T \|\dot{x}(t) - f_0\big(x(t)\big)\|^2, \\ &\qquad + \big\langle C\phi\big(x(t)\big), C\phi\big(x(t)\big)\big\rangle - 2\big\langle \dot{x}(t) - f_0\big(x(t)\big), C\phi\big(x(t)\big)\big\rangle dt + \frac{\lambda}{2}\langle C, C\rangle \end{aligned}$$

and

$$\frac{\partial \mathcal{J}_T(C)}{\partial C} = \frac{1}{2T}\int_0^T 2C\big[\phi\big(x(t)\big) \otimes \phi\big(x(t)\big)\big] - 2\big[\big(\dot{x}(t) - f_0\big(x(t)\big)\big) \otimes \phi\big(x(t)\big)\big]dt + \lambda C.$$

By setting the gradient to zero, we see that

$$C\left[\frac{1}{T}\int_0^T \big[\phi\big(x(t)\big) \otimes \phi\big(x(t)\big)\big]dt + \lambda I\right] = \frac{1}{T}\int_0^T \big[\big(\dot{x}(t) - f_0\big(x(t)\big)\big) \otimes \phi\big(x(t)\big)\big]dt.$$

Finally, we can take the transpose of both sides, apply our definitions of Y, Z, and use symmetry of Z to get

$$[Z + \lambda I]C^T = Y.$$

Chapter 5

LEARNING ABSORPTION RATES IN GLUCOSE-INSULIN DYNAMICS FROM MEAL COVARIATES

Remark 5.0.1. This chapter is derived from the manuscript by Wang, Levine, Shi, and Fox [425], which was published and spotlighted at NeurIPS Timeseries for Health 2022 Workshop.

5.1 Abstract

Traditional models of glucose-insulin dynamics rely on heuristic parameterizations chosen to fit observations within a laboratory setting. However, these models cannot describe glucose dynamics in daily life. One source of failure is in their descriptions of glucose absorption rates after meal events. A meal's macronutritional content has nuanced effects on the absorption profile, which is difficult to model mechanistically. In this paper, we propose to learn the effects of macronutrition content from glucose-insulin data and meal covariates. Given macronutrition information and meal times, we use a neural network to predict an individual's glucose absorption rate. We use this neural rate function as the control function in a differential equation of glucose dynamics, enabling end-to-end training. On simulated data, our approach is able to closely approximate true absorption rates, resulting in better forecast than heuristic parameterizations, despite only observing glucose, insulin, and macronutritional information. Our work readily generalizes to meal events with higher-dimensional covariates, such as images, setting the stage for glucose dynamics models that are personalized to each individual's daily life.

5.2 Introduction

Type-1 diabetes is a chronic condition of glucose dysregulation that affects 9 million people around the world. Decades of research have produced dozens of glucose-insulin dynamics models in order to understand the condition and help diabetics manage their daily lives. These models are typically developed using physiological knowledge and validated in laboratory settings. However, these mechanistic models are incomplete; they are not flexible enough to fit observations outside of controlled settings, due to unmodelled variables, unmodelled dynamics, and external influences. As a result, these mechanistic models fail to fully describe an individual's glycemic response to external inputs like nutrition.

Standard models, such as Dalla Man, Rizza, and Cobelli [99], focus on the glycemic impact of carbohydrates in a meal—carbohydrates are broken down into glucose molecules, then absorbed into blood. However, these models typically ignore other macronutrients, such as fat, fiber, and protein, which are known to contribute substantially to the amount and timing of glucose absorption into the blood. Indeed, this phenomenon is the basis for the glycemic index of various foods. In reality, individual glycemic responses to nutrition go beyond such a simple characterization. For example, Zeevi et al. [451] identified multiple patient sub-groups with different glycemic responses to complex foods.

In our paper, we propose a method that can leverage real-world nutrition and glucose-insulin measurements to improve the fidelity of existing mechanistic models. While we tailor this approach to the specific application of type-1 diabetes, we note that our methodology fits within a broad paradigm of hybrid modeling of dynamical systems [435, 237, 288, 349]. These approaches can improve mechanistic ODEs using flexible components that learn from observations of the system and its external controls.

5.3 Background on modelling glucose-insulin dynamics

Our paper builds on the tradition of modelling physiological dynamics via ordinary differential equations (ODEs), [34, 397, 99, 181, 279]. Traditional models consider ODEs of the form $\dot{x}(t) = f(t, x(t)) + u(t)$, where $x \in \mathbb{R}^n$ denotes physiologic states, $f : \mathbb{R}^n \to \mathbb{R}^n$ encodes mechanistic knowledge of their interactions, and $u : \mathbb{R} \to \mathbb{R}^n$ represents external time-varying inputs into the system. Significant effort has gone towards identifying u from insulin, exercise, and meal data, but u is typically represented via a gastrointestinal ODE model [122, 97] or via hand-chosen functional forms [182, 174, 257]. Both approaches for representing meals depend only on carbohydrate consumption and do not consider other macronutrient quantities.

Our paper considers the minimal model of glucose-insulin dynamics by Bergman et al. [34]:

$$\dot{G}(t) = -c_1[G(t) - G_b] - G(t)X(t) + u_G(t) \tag{5.1a}$$

$$\dot{X}(t) = -c_2 X(t) + c_3[I(t) - I_b] \tag{5.1b}$$

$$\dot{I}(t) = -c_4[I(t) - I_b] + u_I(t) \tag{5.1c}$$

where $x = (G, X, I)$ and $u = (u_G, u_I)$. Here, $G : \mathbb{R} \to \mathbb{R}$ represents plasma glucose concentration, $I : \mathbb{R} \to \mathbb{R}$ represents plasma insulin concentration, $X : \mathbb{R} \to \mathbb{R}$ represents the effect of insulin on glucose, $G_b, I_b \in \mathbb{R}$ represent basal glucose and insulin levels, respectively, and $c_1, c_2, c_3, c_4 \in \mathbb{R}$ represent rate constants for the interactions. Importantly, $u_G : \mathbb{R} \to \mathbb{R}$ represents the appearance of glucose in the blood (e.g. absorbed from nutrition in the gut) and $u_I : \mathbb{R} \to \mathbb{R}$ represents the appearance of insulin in the blood (e.g. absorbed from subcutaneous injection or drip). See Gallardo-Hernández et al. [140] for a modern exposition and the units of each quantity.

Modelling nutrition absorption from discrete meal events. When simulating the daily management of diabetes, the *continuous* functions u_G, u_I are typically derived from observed *discrete* events (e.g. meals and insulin injections). Each discrete-time event $e_i = (t_i, m_i)$ consists of a timestamp t_i and a covariate m_i. If e_i is a meal event, m_i may consist of macronutritional information, an image of the food, or both. Pharmacodynamics models are often used to map the insulin dose to a continuous absorption profile u_I that is compatible with the above model. However, the dependence of glucose absorption u_G on full macronutritional content of a meal event is less well-understood; thus *we focus on modelling u_G in this paper*.

Mechanistic u_G models often derive u_G as the solution to another set of heuristic ODEs[99]. However, this approach introduces additional handcrafted parameterizations to explain quantities that are unobservable outside of the lab setting, such as the glucose concentration in the stomach over time after a meal. A simpler yet effective approach is to directly model u_G phenomenologically, and estimate it from data [174, 257]. Instead of deriving u_G from an intricate model of the human body, this approach represents u_G directly using a parametric function adapted from data.

5.4 Phenomenologically modelling the absorption rate

Let each meal event i be $e_i = (t_i, m_i)$ where $t_i \in \mathbb{R}$ is the meal time and $m_i \in \mathbb{R}^M$ is a vector of meal covariates, such as its macronutrition content or even a photo of the food. We assume we have data on a set E of these meal events. For each meal i, we associate a parametric function $a_i : \mathbb{R}_+ \to \mathbb{R}_+$, such that $a_i(t)$ is the absorption rate of the meal at time t. The overall control function u_G is then a sum over the events:

$$u_G(t) = \sum_{i=1}^{|E|} a_i(t). \tag{5.2}$$

a_i is usually compactly supported, since meals only affects glucose locally in time. Decomposing u_G into a sum allows us to model the effect of each meal individually, instead of all at once.

A simple heuristic choice is a square function $a_i(t) = g_i \mathbb{1}_{[0,w)}(t - t_i)/w$ where w is the width of the square as a free parameter and $g_i \in \mathbb{R}$ is the amount of glucose produced from the meal. Another choice is the bump function $a_i(t) = g_i \mathbb{1}_{[0,\infty)}(t - t_i)(e^{-b_1(t-t_i)} - e^{-b_2(t-t_i)})/b_3$ where b_1 and b_2 are free parameters and b_3 is a normalization constant [10, 389]. For both choices, g_i must be estimated by the patient or by a nutritionist (e.g. when m_i is a food image), which can be highly inaccurate. More importantly, the *shape* of these parameterizations does not depend on m_i, even though foods vary in absorption profiles.

A neural phenomenological model. The form of Equation (5.2) suggests a natural extension that takes advantage of the flexiblity of neural networks. Given a meal event $e_i = (t_i, m_i)$, we model its absorption rate using a neural network a_θ such that

$$a_i(t) = g_i \cdot a_\theta(t - t_i, m_i)\mathbb{1}_{[0,\infty)}(t - t_i). \tag{5.3}$$

We make use of the estimated glucose content g_i following prior approaches since it is often already available in the meals dataset, and gives an expert-informed glucose absorption scale factor. Alternatively, g_i can be included as another input to a_θ instead of being a multiplicative constant. Even if the estimated g_i is inaccurate, a_θ has the flexiblity to rescale g_i based on the observed m_i. Most importantly, our parameterization differs in that its *shape* can adapt to the meal covariates m_i. We share one neural network a_θ across all meal events, allowing it to generalize to macronutritional information similar to, but not exactly the same as, meals from the training set. Altogether, Equations (5.1),(5.2),(5.3) define our neural differential equation model.

End-to-end training on partial observations. Having defined our parametric function, we now discuss how to learn the parameters θ in a setting that is realistic to settings outside of the laboratory. Recent technologies like continuous glucose monitors and artificial pancreases enable real-time measurements of glucose levels and insulin dosage. However, most of a patient's physiological state is unobserved. Within Equation (5.1), we do not observe insulin I and its effect X.

Let x be the state of our differential equation from Equation (5.1). We assume our temporal data consists of noisy partial observations over time $\{(t_k, y_k)\}_{k=1}^T$, where $y_k = Hx(t_k) + \varepsilon$. We assume the projection operator $H : (G, X, I) \mapsto (G, 0, 0)$ and

ε is a zero-mean i.i.d. noise process. Given initial condition $x(t_0) = x_0$, we can numerically integrate Equation (5.1) with a given u_I and our parameterized $u_G(\cdot\,;\theta)$ to obtain an estimate $\hat{y}(t_k) = H\hat{x}(t_k)$ where $\hat{x}(t_k) = \text{Integrate}(f, u, x_0, t_0, t_k)$. We then minimize the mean squared error objective $L(\theta) = \sum_{k=1}^{T} \|\hat{y}(t_k) - y_k\|_2^2 / T$ with respect to θ to fit our parametric model [134]. However, this procedure requires us to know x_0, which is not fully observed in practice.

Many methods exist for performing such under-determined state and parameter estimation; often, the state-estimation component is performed using filtering or smoothing [48, 237, 82, 79, 359], but can also be learnt through other data-driven [23, 205] or gradient-descent [304] methods. In our experiments, we estimate an initial state x_0 by using a sequence of F observations $(G(t_{-F+1}), G(t_{-F+2}), \ldots, G(t_0))$ as a forcing function when forward integrating Equation (5.1), described in Section 4.4.3.3. This simple procedure was sufficient for our model to learn a good θ, likely due to the rapidly decaying autocorrelation of (5.1).

5.5 Experiments

We evaluate our proposed method on simulated data. We simulate 28 days worth of glucose, insulin, and meal data for one virtual patient using Equation (5.1). We evaluate our method against baseline methods with and without glucose observation noise. We also evaluate each method in the realistic setting where the *time* of each meal is noisily reported, since in daily life, the recorded meal time is often only approximately correct.

Data generation. For each day, we generate four meals: breakfast, lunch, dinner, and a late snack. Meals occur uniformly at random within 6-9AM, 11AM-2:30PM, 5-8PM, and 10-11PM, respectively. Each meal contains a glucose amount uniformly random within 5-65g, 20-70g, 40-100g, and 5-15g respectively. For each meal event i, we convert grams of glucose to plasma glucose concentration, assuming the individual has 50dl of blood, and use the result as g_i. To simulate different absorption profiles, each meal is a convex mixture of three "absorption templates". Each template j is given by delayed bump function $a^j(t) \propto g_i \mathbb{1}_{[0,\infty)}(t - t_i - d)(e^{-b_1(t-t_i-d)} - e^{-b_2(t-t_i-d)})$, each with its own set of parameters $(b_1, b_2, d) \in \{(0.04, 0.09, 5\text{min}), (0.08, 0.13, 5\text{min}), (0.03, 0.04, 30\text{min})\}$, visualized in Figure 5.1. The templates represent regular absorption, fast absorption, and slow absorption, respectively. The macronutrition of meal i is then the vector of mixture coefficients $m_i \in \mathbb{R}^3$ such that meal i has absorption profile

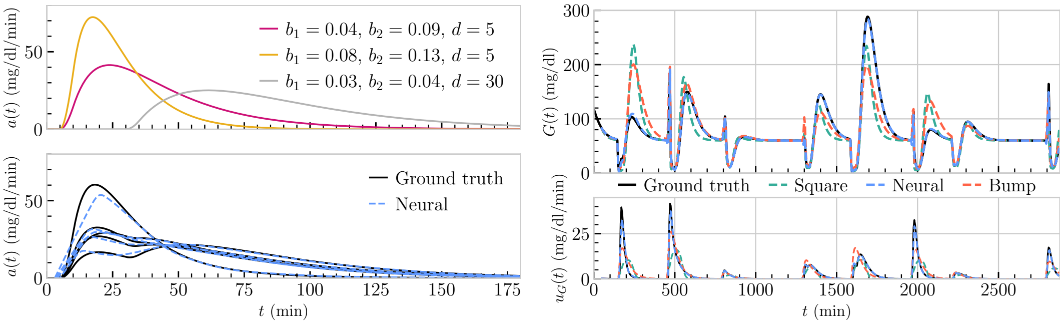


Figure 5.1: *Left*: (Top) “Absorption templates” used to generate u_G. (Bottom) 5 Samples of ground truth and learned u_G for meals from the test set. *Right*: Glucose forecast and predicted absorption rates of each model over a 2 day window from the test trajectory.

$a_i(t) = \sum_{j=1}^{3} m_{ij} a^j(t)$. To ensure a_i is smooth, we average each value $a_i(t)$ with a grid of 50 points from the past 5 minutes.

For each meal time t_i, we simulate an insulin bolus dose at a time sampled from $\mathcal{N}(t_i, (10\text{min})^2)$. We sample a glucose to insulin conversion for each meal from $\mathcal{N}(7\text{g/U}, (1\text{g/U})^2)$. To simulate imperfect measurements, we add a relative $N(0, 0.05^2)$ observation noise. To simulate imperfect meal time recordings, we add $N(5\text{min}, (2.5\text{min})^2)$ noise to meal times. We use a square function u_I, corresponding to a constant insulin absorption rate, over 30 minutes, which we assume to be known to every model. We use parameters from Andersen and Højbjerre [18] for Equation 5.1, and we use Euler integration with a step size of 0.1 minutes to produce an observation every 5 minutes.

Experimental setup. We split our generated data temporally into 3 disjoint training, validation, and testing trajectories. We optimize using Adam [208] for 1000 iterations, with a half-period cosine learning rate schedule following a linear ramp up to 0.2 over the first 30 iterations. We use minibatches of 512 sequences of 4 hour windows (48 observations) and use 10 observations for estimating the initial condition. We minimize the mean squared error on the observed glucose values with respect to the parameters θ of a_θ, keeping the other parameters of Equation (5.1) fixed. We parameterize our neural a_θ using a feedforward network with 2 hidden layers of 64 units and GELU activations. We found that appropriately scaling the input and outputs of a_i is crucial for stable optimization.

Evaluations. We compare our neural absorption function against the two common parameterizations of u_G from Section 5.4, fit via gradient-based optimization.

	Exact timestamps		Noisy timestamps	
a_i	Exact observations	Noisy observations	Exact observations	Noisy observations
Neural	0.95mg/dl	3.66mg/dl	1.48mg/dl	3.63mg/dl
Bump	9.52mg/dl	10.11mg/dl	9.53mg/dl	10.24mg/dl
Square	11.60mg/dl	11.53mg/dl	11.65mg/dl	11.56mg/dl

Table 5.1: Forecast RMSE computed over all possible 4 hour windows of the test set trajectory, reflecting the window size used for training.

We approximate the piece-wise constant square function using a difference of sigmoids; otherwise the width cannot be learned. Our neural model is able to closely approximate the ground truth u_G, especially in the tails, as shown in Figure 5.1 (left). This results in significantly better forecasts, and our neural model closely tracks the ground truth glucose values and absorption rates, *even extrapolating to durations much longer than what was seen in training*. We visualize such long term forecasts in Figure 5.1 (right). We also report the forecast RMSE on the test set in Table 5.1. Our neural model attains lower forecast errors across all settings. In the noiseless case, our neural model is 10x more accurate than heuristic parameterizations. The RMSEs generally increase as we add noise, though the bump and square functions are already such poor forecasters that noise does not worsen their errors significantly.

5.6 Discussion

Our experiments show that our proposed method is a promising way to learn absorption profiles that depend on macronutritional information. Our approach readily generalizes to handle arbitrary meal covariates beyond macronutritional information, such as food images or descriptions. Although this paper only uses synthetic data, our method can complement any glucose dynamics model of real-world data. Learning accurate dynamics from data, however, remains a challenging problem. We see our method as a vital component in future data-driven hybrid models of glucose-insulin dynamics.

Chapter 6

LEARNING ABOUT STRUCTURAL ERRORS IN MODELS OF COMPLEX DYNAMICAL SYSTEMS

Remark 6.0.1. This chapter is derived from the manuscript in preparation by Huang, Levine, Schneider, Shen, Stuart, and Wu [187].

6.1 Introduction

Numerical simulation is at the heart of modeling, predicting, and understanding dynamical systems that are too complex to be amenable to analytical solution. Complex dynamical systems here extend from molecular dynamics with quantum effects to the planetary scales of weather and climate. The range of dynamically important scales in these systems can be vast, for example, in case of the atmosphere, extending over 13 orders of magnitude from the micrometers of cloud droplets and aerosols to the tens of thousands kilometers of planetary waves. The number of degrees of freedom that would need to be resolved for a faithful simulation of such systems (e.g., $\gtrsim 10^{21}$ for a typical atmospheric boundary layer flow) often exceeds what will be computationally feasible for the foreseeable future [377].

Instead of direct numerical simulation, a variety of approaches have been devised to approximately resolve the most important degrees of freedom in numerical simulations. The degrees of freedom that remain unresolved but, because of nonlinear interactions, are still important for the resolved degrees of freedom are then represented by closure models, which link what is unresolved to what is resolved. The state X of the approximate system evolves according to dynamics of the form

$$\dot{X} = f(X; \theta_{\mathrm{P}}), \tag{6.1}$$

where f may depend on derivatives of the state X; hence, the system may represent partial differential equations. The system depends on empirical parameters θ_{P} that appear in closure models. For example, in large-eddy simulations of turbulent flows, the most energetic “large eddies” are explicitly resolved in the dynamics represented by f. The effect of the unresolved scales is modeled by subgrid-scale models, such as the classical Smagorinsky model [390], which depend on empirical parameters θ_{P} (e.g., the Smagorinsky coefficient). Similar semi-empirical models are used in many other fields. They encode domain-specific knowledge, and their parameters θ_P need to be calibrated with data.

Data y that are informative about the system come in a variety of forms, such as direct measurements of the time evolution of the state X or more indirect mappings of the state X onto observables, which may, for example, be statistical aggregates of state variables or convolutions of state variables with kernels. Convolutional data arise, for example, when representing the effect of a state variable such as temperature on the radiative energy fluxes that a satellite measures from space. Generally, we can write that the state maps to observables via an observation operator $\mathcal{H}$, such that

$$\hat{y} = \mathcal{H}(X). \tag{6.2}$$

The challenge is that simulated observables $\hat{y}$ generally are biased estimates of actual data y. The actual data y are affected by measurement error, and the simulated data $\hat{y}$ are affected by structural errors in the approximate dynamical system (6.1). For example, while a general feature of turbulence is to enhance mixing of conserved quantities, turbulent mixing is not always diffusive in character. Therefore, diffusive subgrid-scale models such as the Smagorinsky model are not always structurally correct, especially in convective situations with coherent flow structures [179]. This can lead to biases that, for example, adversely affect the calibration of model parameters θ_P.

The purpose of this paper is to summarize principles of, and algorithms for, learning about structural error models that correct semi-empirical closure models. Wholesale replacement of semi-empirical closure models with neural networks and other deep learning approaches promises to overcome the structural strictures of existing closure models through more expressive models; it has recently received much attention [254, 430, 343, 283, 118, 459, 372, 60]. However, existing semi-empirical closure models encode valuable domain-specific knowledge. Learning flexible corrections to these models is often less data hungry, more interpretable, and potentially more generalizable than replacing them wholesale.

What follows is a distillation of experiences we gained in studying various complex dynamical systems. Our goal is to provide guidelines and algorithms that can lead to a broad-purpose computational framework for systematically learning about model error in dynamical systems. We focus two important parts of error modeling: (i) how to construct an error model, and (ii) how to calibrate an error model.

Our approach to constructing error models builds upon but goes beyond the classical work of Kennedy and O'Hagan [206], who accounted for model error through an external bias correction term $\delta(X;\theta_{\mathrm{E}})$, parameterized by parameters θ_{E} and inserted at the boundary between output from a computer model $\hat{y} = \mathcal{G}(\theta_{\mathrm{P}})$ and data y:

$$y = \hat{y} + \delta(X;\theta_{\mathrm{E}}) + \eta. \tag{6.3}$$

Here, $\mathcal{G}(\theta_{\mathrm{P}}) = \mathcal{H}[X(\theta_{\mathrm{P}})]$ corresponds to solving (6.1) for the time series of the state X, which depends parameterically on θ_{P}, and then applying the observation operator (6.2); hence, $\mathcal{G}$ is a mapping from the space of model parameters θ_{P} to the space of observations y. The noise η represents additional (e.g., observation) errors, assumed to have zero mean. In this approach, the model parameters θ_{P} remain fixed (i.e., a property of $\mathcal{G}$) while parameters θ_{E} in the error model δ are tuned such that the residual $y - \hat{y}$ has a small magnitude and zero mean. This approach of externalizing model error for bias correction has been applied and further expanded in many subsequent papers [e.g., 175, 428, 63, 409, 69]. A key advantage of the external model error approach is that the model producing $\hat{y}$ can be treated as a black-box, which facilitates use of this approach across different domains. State variables X and residuals $y - \hat{y}$ form input-output pairs from which the error model $\delta(X;\theta_{\mathrm{E}})$ can be learned, for example, with supervised learning[1] approaches, usually in an offline setting separate from learning about the model parameters θ_{P}.

However, the external model error approach has several drawbacks:

- It is difficult to incorporate physical (or other process-based) knowledge or constraints (e.g., conservation laws) in the error model $\delta(X;\theta_{\mathrm{E}})$ [63].
- It cannot improve predictions for quantities other than the observations y on which the error model $\delta(X;\theta_{\mathrm{E}})$ has been trained.
- It leads to interpretability challenges because $\delta(X;\theta_{\mathrm{E}})$ is a catch-all error term that typically represents the sum total of errors made in several, and often disparate, closure models.

To address these drawbacks, a few researchers have started to explore an approach that internalizes model error [394, 171, 362]. Such an internal model error approach embeds L error models $\delta_l(X;\ \theta_{\mathrm{I}}^{(l)})$ $(l = 1, \ldots, L)$ within the dynamical system, at

[1] Supervised learning refers to regression or interpolation of data.

the places (e.g., in closure models) where the errors are actually made. Let their collection be written as

$$\delta(X;\ \theta_{\mathrm{I}}) := \left\{\delta_l\left(X\ ;\ \theta_{\mathrm{I}}^{(l)}\right)\right\}_{l=1}^{L} \tag{6.4}$$

so that we can write for the overall system

$$\dot{X} = f\Big(X;\ \theta_{\mathrm{P}}, \delta(X;\ \theta_{\mathrm{I}})\Big). \tag{6.5}$$

The error models internal to the dynamical system are chosen so that the error-corrected computer model $\hat{y} = \mathcal{G}(\theta_{\mathrm{P}};\ \theta_{\mathrm{I}}) = \mathcal{H}[X(\theta_{\mathrm{P}};\ \theta_{\mathrm{I}})]$ provides unbiased estimates of the data y:

$$y = \mathcal{G}(\theta_{\mathrm{P}};\ \theta_{\mathrm{I}}) + \eta, \tag{6.6}$$

where the additional errors η are still assumed to have zero mean. Figure 6.1 illustrates and contrasts the external and internal approaches to modeling structural errors.

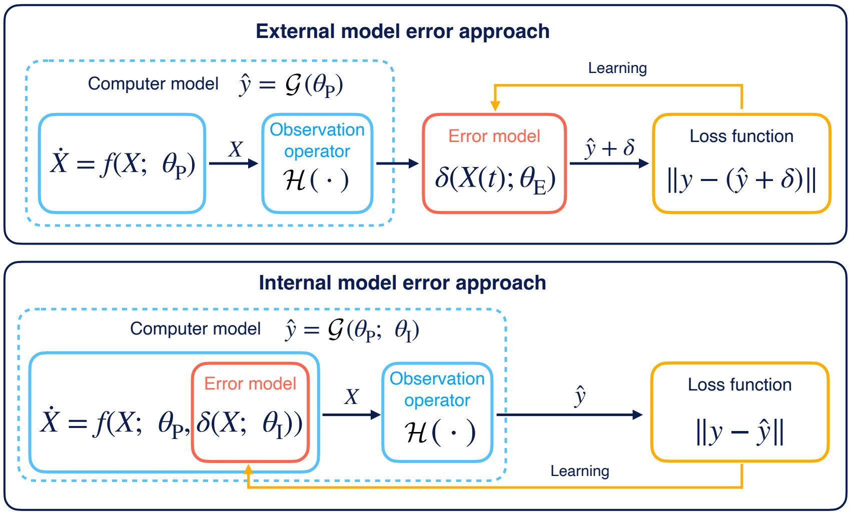

External model error approach
Computer model $\hat{y} = \mathcal{G}(\theta_{\mathrm{P}})$
Learning
$\dot{X} = f(X;\ \theta_{\mathrm{P}})$
X
Observation operator
$\mathcal{H}(\cdot)$
Error model
$\delta(X(t); \theta_{\mathrm{E}})$
$\hat{y} + \delta$
Loss function
$\|y - (\hat{y} + \delta)\|$
Internal model error approach
Computer model $\hat{y} = \mathcal{G}(\theta_{\mathrm{P}};\ \theta_{\mathrm{I}})$
Error model
$\dot{X} = f(X;\ \theta_{\mathrm{P}}, \delta(X;\ \theta_{\mathrm{I}}))$
X
Observation operator
$\mathcal{H}(\cdot)$
$\hat{y}$
Loss function
$\|y - \hat{y}\|$
Learning

Figure 6.1: External and internal approaches to modeling structural errors in complex dynamical systems.

Such an approach has found applications, for example, in turbulence modeling [123, 303, 85, 442, 321, 424, 440]. By incorporating the structural error models $\delta(X;\ \theta_{\mathrm{I}})$ inside the dynamical system, the error models can in principle lead to improved predictions even of quantities that were not used to train the error models. The error models can be learned alongside the model parameters θ_{P} in an online setting. They also are more amenable to interpretation because they are included in the places where errors are actually made. A potential downside of internalizing model error is that the effects of the structural errors δ map onto data y only through the solved state X of the dynamical system, and residuals $y - \hat{y}$ are generally not directly informative about how the structural errors $\delta(X;\ \theta_{\mathrm{I}})$ depend on state variables X; thus, learning about structural errors $\delta(X;\ \theta_{\mathrm{I}})$ can generally not be accomplished with direct supervised learning approaches. Instead, residuals $y - \hat{y}$ only provide indirect information about structural errors $\delta(X;\ \theta_{\mathrm{I}})$. Additionally, if derivatives of the dynamical system f with respect to parameters are not easily available, or if the dynamical system is not differentiable, gradient-based methods for learning about the model errors $\delta(X;\ \theta_{\mathrm{I}})$ are difficult or impossible to use.

Here we show various ways of constructing models for structural errors and demonstrate how one can learn about the structural errors from direct or indirect data in the absence of derivatives of the dynamical system. As error models, we will consider:

- Gaussian processes, as in Kennedy and O'Hagan [206];
- Models assembled from dictionaries of terms (e.g., involving differential operators), as in the data-driven discovery of partial differential equations [366, 360, 365, 448, 385, 234];
- Neural networks, for their expressivity [254, 55, 343, 283, 449, 56];
- Stochastic models, because without a clear scale separation between resolved and unresolved degrees of freedom, homogenization theory suggests that closure models generally should be stochastic [e.g., 465, 271, 136, 306];
- Non-local models, because structural errors may be non-local in space [460, 463, 115, 96, 307], in time [273, 426, 252], or in both [73].

We will discuss how to learn about such error models both from direct and indirect data. Supervised learning of various types of error models δ_l from direct data has been performed in a number of settings, for example, to discover terms in differential equations or neural network closure models from residual time tendencies that give direct information about the error model that is sought [e.g., 59, 424, 440, 236]. However, data directly informative about error models, such as high-resolution time tendencies, are not always available. When training an error model on time tendencies, it can also be difficult to ensure both stability of the dynamical system including the error model and satisfaction of physical constraints (e.g., energy conservation) [e.g., 56]. Training an error model with indirect data, in an inverse problem rather than supervised learning setting [e.g., 442, 376, 456], can be advantageous in those circumstances. We demonstrate how it can be accomplished.

The principles and algorithms we will discuss are broad purpose and applicable across a range of domains and model complexities. To illustrate their properties in a relatively simple setting, we use two versions of the Lorenz 96 [261] dynamical system: the basic version of the model and its multiscale generalization [133]. Section 6.2 discusses the calibration of internal error models with direct or indirect data and the enforcement of constraints such as conservation properties. Section 6.3 introduces various ways of constructing error models. Section 6.4 introduces two Lorenz 96 systems and then proceeds to present various concepts and methods through numerical results for these systems. Section 6.5 is organized similarly to the previous section, but concerns a model of the human glucose-insulin system. Section 6.6 summarizes the conclusions.

6.2 Calibrating Error Models

We first summarize several important aspects of calibrating internal error models, including (i) direct or indirect data, (ii) gradient-based or derivative-free optimization, and (iii) enforcing constraints (e.g., sparsity or physical laws). We discuss these aspects with a generic internal error model $\delta(X; \theta_{\mathrm{I}})$, which may represent any one of the error model types we will introduce later.

6.2.1 Data Availability

6.2.1.1 Direct Data

Direct data to calibrate δ are defined as "labeled" input-output pairs $\{X(t_i), \delta(X(t_i))\}_{i=1}^{N}$, where i denotes a time index. Consider the additive error model

$$\dot{X} = f(X; \theta_{\mathrm{P}}) + \delta(X; \theta_{\mathrm{I}})$$

as an example. A fine temporal resolution of $X(t)$ is usually needed to approximate $\dot{X} - f(X; \theta_{\mathrm{P}})$ and obtain estimates of the error $\delta(X; \theta_{\mathrm{I}})$ as a residual. With this method, it becomes challenging to obtain reliable direct data when the trajectories $\dot{X}$ are noisy, for example, when the dynamical system is chaotic [59]. Furthermore it may not be possible to observe the entirety of X. This may be handled as a missing data problem [256], and could be handled by joint parameter-state estimation for example using data assimilation; see [42, 83].

An additional complication with using direct data is ensuring the stability of the dynamical system with the calibrated error model $\delta(\cdot)$. Although with direct data, we can get more control of the accuracy of the error model itself, the calibrated error model often leads to unstable simulations of the dynamical system with the error model [29, 54]. There are several ways to mitigate the instability introduced by the error model, e.g., adopting a structure that ensures physical constraints [447], enforcing physical constraints [38], ensuring stability by bounding the eigenvalues of the linearized operator, and limiting the Lipschitz constant of $\delta(\cdot)$ [383]. However, a systematic approach to ensure stability is lacking.

6.2.1.2 Indirect Data

Instead of assuming access to direct data $\{X, \delta(X)\}$, the error model can also be calibrated with indirect data by solving an inverse problem associated with (6.32) (i.e., solve for the most likely parameters $\theta_{\mathrm{P}}, \theta_{\mathrm{I}}$ given the model $\mathcal{G}$ and data y). Using indirect data involves simulating the dynamical system with the error model as in (6.5); therefore, the calibration procedure with indirect data favors error models that lead to stable simulations, an important advantage over the direct methods. Typical examples of problems giving rise to indirect data include time-series of X for which the resolution is not fine enough to extract direct data for calibration [47], time-averaged statistics of X [373] or dynamical systems which are partially observed [325]. More generally, indirect data can also be interpreted as constraints on X, and thus physical constraints can be enforced via augmenting indirect data.

6.2.2 Methods of Calibration

Using direct data $\{X, \delta(X)\}$ for calibration leads to a regression problem, which can be solved with standard methods for a given parameterization of the error model (e.g., least squares fit for dictionary learning, gradient descent methods for neural network). By contrast, using indirect data for calibration leads to an inverse problem associated with Eq. (6.32). Indirect methods can be linked to direct methods by framing as a missing data problem [256] and alternating between updating the missing data and then updating the calibration parameters using learned direct data, for example using the EM algorithm [285]. However, in this section we focus on the calibration in the inverse problem setting, without introducing missing data, and discussing gradient-based and derivative-free optimization methods and how to enforce constraints.

6.2.2.1 Gradient-based or Derivative-free Optimization

Eq. (6.32) defines a forward problem in which $\mathcal{G}$, θ_{P}, θ_{I} and noise η can be used to generate simulated data. The associated inverse problem involves identifying the most likely parameters $\theta_{\mathrm{P}}, \theta_{\mathrm{I}}$ for $\mathcal{G}$, conditioned on observed data y. To formalize this, we first define a loss function

$$\mathcal{L}(\theta) = \frac{1}{2} \left| y - \mathcal{G}\left(\theta_{\mathrm{P}};\ \theta_{\mathrm{I}}\right)\right|_{\Sigma}^{2}, \tag{6.7}$$

where $\theta = [\theta_{\mathrm{I}}, \theta_{\mathrm{P}}]$ and Σ denotes the covariance of the zero-mean noise η.[2] The inverse problem

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\theta)$$

can be solved by gradient descent methods once the gradient

$$\frac{d\mathcal{L}}{d\theta} = \frac{d\mathcal{G}}{d\theta}^{T} \Sigma^{-1}(y - \mathcal{G}) \tag{6.8}$$

is calculated. In practice, the action of the term $d\mathcal{G}/d\theta^T$ is often evaluated via adjoint methods for efficiency. Although the gradient-based optimization is usually more efficient when $\mathcal{G}$ is differentiable, the evaluation of $\mathcal{G}$ can be noisy (e.g., when using finite-time averages to approximate infinite-time averaged data [116]) or stochastic (e.g., when using stochastic processes to construct the error model). In these settings, gradient-based optimization may no longer be suitable, and derivative-free optimization becomes necessary. In this paper we focus on Kalman-based derivative-free optimization for solving the inverse problem; 6.7 briefly reviews a specific easily implementable form of ensemble Kalman inversion (EKI), to illustrate how the methodology works, and gives pointers to the broader literature in the field.

6.2.2.2 Enforcing Constraints

There are various types of constraints that can be enforced when calibrating an error model. Two most common constraints are sparsity constraints and physical constraints (e.g., conservation laws). Here we present the general concept of enforcing these two types of constraints in calibration, and 6.7 presents more details about using EKI to solve the corresponding constrained optimization problems.

[2] By $|\cdot|_B$, we denote the covariance-weighted norm defined by $|v|_B = v^* B^{-1} v$ for any positive-definite B.

To impose sparsity on the solution of θ_{I}, we aim to solve the optimization problem [374]

$$\begin{aligned} \mathcal{L}(\theta;\lambda) &:= \frac{1}{2}\left|y - \mathcal{G}\left(\theta_{\mathrm{P}};\ \theta_{\mathrm{I}}\right)\right|_{\Sigma}^{2} + \lambda|\theta_{\mathrm{I}}|_{\ell_0}, \\ \theta^{*} &= \underset{\theta\in\mathcal{V}}{\arg\min}\, \mathcal{L}(\theta;\lambda), \end{aligned} \tag{6.9}$$

where $\mathcal{V} = \{\theta : |\theta_{\mathrm{I}}|_{\ell_1} \leq \gamma\}$. The regularization parameters γ and λ can be determined via cross-validation. In practice, adding the ℓ_0 constraint is achieved by thresholding the results from ℓ_1-constrained optimization. The detailed algorithm was proposed in [374] and is summarized in 6.7.

In many applications, we are also interested in finding a solution of θ_{I} that satisfies certain physical constraints, e.g., energy conservation. To impose physical constraints on the solution of θ_{I} from EKI, we first generalize the constraint as:

$$\mathcal{V} = \{\theta : \mathcal{R}(\theta_{\mathrm{I}}) \leq \gamma\}. \tag{6.10}$$

Here, $\mathcal{R}$ can be interpreted as a function that evaluates the residuals of certain physical constraints (typically, by solving Eq (6.5)). The constraint function $\mathcal{R}$ can be nonlinear with respect to θ_{I}. Taking the additive error model $\dot{X} = f(X) + \delta(X;\theta_{\mathrm{I}})$ as an example, the function $\mathcal{R}$ corresponding to the energy conservation constraint can be written explicitly as

$$\mathcal{R}(\theta_{\mathrm{I}}) = \left|\int_0^T \big(\langle\delta(X(t);\theta_{\mathrm{I}}), X(t)\rangle\big)dt\right|, \tag{6.11}$$

which constrains the total energy introduced into the system during the time interval $[0,T]$. Alternatively, a stronger constraint can be formulated as

$$\mathcal{R}(\theta_{\mathrm{I}}) = \int_0^T \big|\langle\delta(X(t);\theta_{\mathrm{I}}), X(t)\rangle\big|dt, \tag{6.12}$$

which constrains the additional energy introduced into the system at every time step within the time interval $[0,T]$. The notation $\langle\cdot,\cdot\rangle$ denotes the inner product. Both forms of constraint in (6.11) and (6.12) can be implemented by using augmented observations, i.e., including the accumulated violation of energy constraint as a additional piece of observation data whose true mean value is set to zero.

6.3 Constructing Error Models

To be concrete we highlight three different approaches to representing structural errors: dictionary learning, Gaussian processes, and neural networks; however the reader may wish to consider the use of other representations for structural error, within the overarching framework proposed here. For these three approaches, existing work mainly focuses on constructing deterministic error models that are locally dependent on state variables; however the approaches can all be extended to the construction of stochastic error models or can be made non-locally dependent on state variables as described in Sections 6.3.4 and 6.3.5. For simplicity, we define the error models for the whole collection of structural errors $\delta(X, \theta_{\mathrm{I}})$ as written in (6.5); however, we can also define and learn them independently for each component of the structural error model $\delta_l(X, \theta_{\mathrm{I}}^l)$ for $l = 1, \ldots, L$.

6.3.1 Dictionary Learning

If a set of candidate terms in error models is known or can be approximated, an error model can be constructed via learning from a dictionary of J candidate terms,

$$\delta(X; \theta_{\mathrm{I}}) = \sum_{j=1}^{J} \alpha_j \phi_j(X; \beta_j), \tag{6.13}$$

where $\theta_{\mathrm{I}} = \{\alpha, \beta\}$ and $\phi_j(X; \beta_j)$ denote user-specified, parametric basis functions that can, for example, include differential operators [59, 360, 366, 365]. In practice, it is difficult to know all suitable basis functions a priori, and thus it is common to include redundant basis functions in the dictionary. Basis functions can then be pruned based on data by imposing sparsity constraints on the coefficients α_j. Such sparsity constraints have proven to be beneficial in the construction of data-driven models of dynamical systems [59, 360, 366, 365, 374]. They are also commonly used in compressed sensing [110], where dictionary learning has been widely used.

An advantage of using dictionary learning is the potential interpretability of the constructed error model, arising because the error model is a linear combination of user-specified and hence interpretable basis functions. On the other hand, this approach can be overly restrictive when the dictionary of basis functions $\{\phi_j\}$ is misspecified, resulting in an insufficiently expressive error model.

6.3.2 Gaussian Processes

Another option of constructing an error model is via Gaussian processes (GPs) [436],[3]

$$\delta(X; \theta_{\mathrm{I}}) \sim \mathcal{GP}\left(m, \mathcal{K}\right), \tag{6.14}$$

[3] We emphasize that here we use only the mean of the GP and the methodology is simply a form of data-adapted regression; at this point we are not utilizing any of the uncertainty quantification that may be used with GPs.

where $m : \mathcal{X} \mapsto \mathbb{R}$ denotes the mean of δ and $\mathcal{K} : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ represents a kernel. Given data at J different points $X^{(j)}$ for $j = 1, 2, ..., J$, the mean of the error model can be written as a linear combination of basis functions,

$$m(X) = \sum_j \alpha_j \mathcal{K}(X^{(j)}, X; \psi), \tag{6.15}$$

where ψ denotes the hyper-parameters of the kernel $\mathcal{K}$. Therefore, the parameters that characterize the error model become $\theta_{\mathrm{I}} = \{\alpha, \psi\}$ if the mean of a GP is used to represent the model error term δ. The GP approach requires the choice of a kernel $\mathcal{K}$, which then determines the kernel functions $\mathcal{K}(X^{(j)}, \cdot)$ in Eq. (6.15). This may appear restrictive, but we highlight the fact that the hyper-parameters of $\mathcal{K}$ are learned from the data; thus the set of functions in which the solution is sought is data-adapted. This confers a potential advantage over dictionary learning, in particular for problems lacking in strong prior knowledge about the functional form of the model $m(\cdot)$ to be learned. In the case of indirect data, the locations $X^{(j)}$ must also be chosen a priori (or learnt as additional parameters).

Because of the similar forms of Eqs. (6.15) and (6.13), the GP shares similar shortcomings as dictionary learning when the kernel $\mathcal{K}$ is misspecified, even in the presence of hyper-parameter learning. In practice, a more sophisticated kernel $\mathcal{K} = \sum_i \mathcal{K}_i$ is often constructed from some basic kernels $\mathcal{K}_i$ [163, 100, 229]. If a redundant set of basic kernels is used, sparsity constraints can be imposed in a similar way as in dictionary learning to prune the kernel set. A further limitation of using GPs is the computational cost, which grows exponentially with the dimension of $\mathcal{X}$. This pathology can be ameliorated by representing the GP as a linear combination of Random Fourier Features [336], which allows us to recast a GP as a dictionary-based approach in which the bases ϕ_j are drawn randomly from a special distribution known to reproduce a kernel of interest.

6.3.3 Neural Networks

Compared to dictionary learning, neural networks are more expressive, and they are more scalable than GPs, as the latter suffer from the curse of dimensionality if the model has high-dimensional input. Neural networks can also be used to construct an error model,

$$\delta(X; \theta_{\mathrm{I}}) = \mathcal{NN}(X; \theta_{\mathrm{I}}), \tag{6.16}$$

where $\mathcal{NN}$ denotes a neural network and θ_{I} the coefficients (biases and weights) of the neural network. While neural networks are expressive and scalable, it is more difficult to enforce stability of a dynamical system with a neural network error model [54]. This is mainly because the nonlinearity introduced by a neural network is often more difficult to analyze compared with dictionary learning, for which we explicitly specify basis functions and thus can avoid using those that lead to instability; it is also more difficult to analyze than GP based learning because the latter is easier to interpret, as the kernels are hand-picked and then tuned to data. In Section 6.2.2.2, we discuss a general approach to enhancing stability by enforcing energy constraints in the context of learning from indirect data.

6.3.4 Stochastic Extension

In the preceding sections, we briefly summarized commonly used tools for constructing error models. All of those models were deterministic, with fixed parameters θ_{I}. To quantify uncertainties, we can take a Bayesian perspective, view the unknown parameters as random variables, and infer the distributions of those parameters given the data. We can then propagate the uncertainties of those parameters to the simulated state X and predicted observations $\hat{y}$ via Monte Carlo simulations. Although this is a standard approach to quantifying uncertainties, it cannot directly account for the impact of neglected information of unresolved scales upon the resolved state X. The randomness of the unresolved state can have an order one impact upon X; this issue is particularly prevalent in applications without a clear scale separation, such as turbulence, but can also happen in scale-separated problems. In such a scenario, directly modeling this impact as randomness in the resolved state becomes more appropriate, and it can be achieved by further adding a stochastic processes to the deterministic error model:

$$\delta\Big(X;\theta_{\mathrm{I}}\Big) = \delta_{\mathrm{det}}(X;\theta_{\mathrm{det}}) + \sqrt{\sigma^2(X;\theta_{\mathrm{ran}})}\dot{W}, \tag{6.17}$$

where det indicates a deterministic model, W denotes the Wiener process and the overall unknown parameters are defined as $\theta_{\mathrm{I}} = \{\theta_{\mathrm{det}}, \theta_{\mathrm{ran}}\}$. In practice, the above formulation can be further generalized by using stochastic processes (e.g., with desired temporal correlations) other than the Wiener process.

Fitting stochastic models to time-series data has been explored in some previous works [433, 354, 353, 309]; a common problem when applying these methods is the inconsistency between data and the incremental structure of the Gaussian noise driving the model as time step is approaching zero [455, 318, 310, 44]. A common practice to address this issue is the multi-scale use of data, e.g., via subsampling [454, 325, 308, 317, 28, 2]. Some previous works also explored Kramers–Moyal averaging with finite sampling rate correction [180, 222, 67]. On the other hand, fitting a discretized version of stochastic processes to time-series data has been explored using autoregressive models [20, 265]. For some dynamical systems, the unresolved state has conditional (with respect to the resolved state) Gaussian statistics [78, 77], and then fitting the stochastic models can be achieved using analytically derived likelihoods.

In the absence of the whole trajectories of time-series data, some recent works started to explore fitting stochastic models to statistics of time-series data [219, 218, 200, 376]. Using time-averaged data to estimate linear SDEs has been studied for decades to account for climate variablity [168, 135, 320], and extension to nonlinear SDEs was discussed in [167]. In addition, fitting discretized versions of stochastic processes with statistics of time-series data has also been explored in [297] using autoregressive models.

6.3.5 Representing Non-local Effects

Spatial Non-locality: The states $X(t)$ for approximate models and their structural corrections typically consider $X(t)$ as a discretized spatial field. Most traditional closure models are formed locally; that is, they rely on the assumption of local dependence on $X(t,r)$, where $X(t,\cdot) : \mathbb{R}^p \mapsto \mathbb{R}$ is a spatial field, and $r \in \mathbb{R}^p$ represents the spatial coordinate. For some applications, it is useful to consider non-local effect in the error model. Indeed, our formulations of the approximate physical model in (6.1) and models for structural error in Sections 6.3.1 to 6.3.3 are well-specified for scalar (local, component-wise) or vector-valued (non-local) X. Moreover, we note that non-local functions of the state $X(t)$ are best conceptualized as function-valued *operators*—while they take as inputs a vector of neighboring coordinates from $X(t)$, this vector represents a discretized spatial function. Thus, when designing spatially non-local closures, it is often sensible to build them to be consistent across different spatial discretizations.

In the case of neural networks, we can build spatially non-local closures with convolutional neural networks (CNNs); the Fourier Neural Operator (FNO) [246] or deep operator network (DeepONet) [269] provide an extension to an operator limit. Similarly, Gaussian Processes (GPs) and Random Feature Methods (a dictionary-based formulation of GPs) can be designed with spatially non-local vectorized inputs from $X(t)$; recent theoretical work has also allowed these basic methods to be taken to a continuous operator limit [296, 80].

As an emerging topic in the context of data-driven modeling, some recent works have explored non-local diffusion [115, 96, 113, 307] and spatially non-local modeling [460, 463, 73]. In this work, we capture the spatially non-local dependence on X via a data-driven convolution kernel:

$$\delta\Big(X(t,r);\theta_{\mathrm{I}}\Big) = \int_{r'\in\Omega} \delta_{\mathrm{loc}}(X(t,r');\theta_{\mathrm{loc}})C(r-r';\theta_{\text{non-loc}})dr' \tag{6.18}$$

where $\Omega \subset \mathbb{R}^p$ represents a subset of $\mathbb{R}^p$ that contains r, and $C:\mathbb{R}^p \mapsto \mathbb{R}$ denotes a convolution kernel with hyper-parameters $\theta_{\text{non-loc}}$. The overall parameterization is defined by $\theta_{\mathrm{I}} = \{\theta_{\mathrm{loc}}, \theta_{\text{non-loc}}\}$, such that the unknown parameters in the local error model δ_{loc} and the convolutional kernel C can be jointly estimated.

Note that hyper-parameters can be made state-dependent: $\theta_{\text{non-loc}}(X(t,r);\kappa)$; then the additional unknowns κ can be learnt, appending it to θ_{I}. Similarly, learning a nonlinear integral kernel has been discussed in [215] and showed as a continuous generalization of the transformer architecture [417]. The form of non-local closure in (6.18) draws inspiration from a series of works about non-local modeling [113], in which δ_{loc} corresponds to a local Laplace operator. Some mathematical foundations of non-local operators and calculus were summarized in [113], and the connection to fractional differential operators was illustrated in [64].

Temporal Non-locality

Non-locality in time – memory – is also important. Generically, any form of variable elimination or coarse-graining results in memory effects which require, at each current point in time, integration of the entire time-history from the initial condition upto the current time [465]. Such memory effects are undesirable as they lead to computational algorithms which scale poorly with respect to length of time-interval. Markovian models which encapsulate memory can be constructed, for example by introducing a recurrent neural network [236], or by the use of delay embedding [363]; such Markovian models are more computational expedient. Temporally non-local modeling has received significant recent attention [273, 426, 73, 252]. If diffusion/advection mechanisms present in the resolved system, memory effects of any state variable at a given point in space would manifest themselves in the state variables of current time at a spatially non-local region around that given point. For this reason non-local models with a flexible enough kernel could potentially be used to capture memory effects, without significantly increasing the computational costs.

6.4 Lorenz 96 Systems as Illustrative Examples

For our simulation studies concerning structural model error in this section we will take two variants on the celebrated Lorenz 96 model [261], described in the Subsection 6.4.1. Then, in Subsection 6.4.2, we present numerical result based on these models.

6.4.1 Lorenz Models Considered

We consider a multiscale Lorenz 96 model, together with a single-scale companion model, to illustrate the principles and algorithms described in the subsequent sections. In each case, we use an untruncated version of the model as the true data-generating model and a truncated version as the model in which structural error models are to be learned.

6.4.1.1 Multiscale Lorenz 96 Model

The Lorenz 96 multi-scale system [261] describes the evolution of a simplified atmospheric flow, which is periodic along latitude circles (space). It does so through one set of slow variables, x_k ($k = 1, \dots, K$), coupled to a set of fast variables, $z_{j,k}$ ($j = 1, \dots, J$), whose indices label space coordinates:

$$\begin{aligned} \dot{x}_k &= -x_{k-1}(x_{k-2} - x_{k+1}) - x_k + F - hc\bar{z}_k, \\ \frac{1}{c}\dot{z}_{j,k} &= -bz_{j+1,k}(z_{j+2,k} - z_{j-1,k}) - z_{j,k} + \frac{h}{J}x_k. \end{aligned} \tag{6.19}$$

Reflecting the periodicity along latitude circles, the variables are periodic in their indices, with

$$x_{k+K} = x_k, \qquad z_{j,k+K} = z_{j,k}, \qquad z_{j+J,k} = z_{j,k+1}. \tag{6.20}$$

The coupling term $hc\bar{z}_k$ describes the impact of the fast dynamics on the slow dynamics, with only the average

$$\bar{z}_k = \frac{1}{J}\sum_{j=1}^{J} z_{j,k} \tag{6.21}$$

of the fast variables affecting the slow variables. To generate data, we work with the parameter choices $K = 36$, $J = 10$, and $F = b = 10$ [261, 373]. The choices of h and c are summarized in Subsection 6.4.2 for different cases.

To study how to model structural errors, we consider a coarse-grained system in which we only simulate approximate versions X_k of the slow variables x_k, neglecting the fast variables. The approximate slow variables are governed by the system,

$$\begin{aligned}\dot{X}_k &= -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F + \delta(X_k, X_k^-; \theta_{\mathrm{I}}),\\ X_{k+K} &= X_k,\end{aligned} \tag{6.22}$$

for $X_k^- = (X_{k-d}, \cdots, X_{k-1}, X_{k+1}, \cdots, X_{k+d})$. Here, $\delta(\cdot)$ is the error model that accounts for the missing multiscale interactions. If there is no dependence on X_k^- we say the model is local; otherwise we allow for non-local dependency with stencil of width d on either side of X_k. If specified correctly, the model error ensures that X_k approximates x_k the solution of (6.22). We will use data generated with the full system (6.19)–(6.21) to learn about the error model $\delta(\cdot)$ in the coarse-grained system (6.22).

As data $y = \mathcal{H}(x(t))$ we consider, let $\mathbb{E}$ denote expectation with respect to the stationary distribution of x. If $x(\cdot) \in \mathcal{X} := C(\mathbb{R}^+; \mathbb{R}^K)$ denotes a solution trajectory of the system and $\mathcal{F} : \mathcal{X} \mapsto \mathbb{R}^q$ is a function on the space of solution trajectories, where q denotes the dimension of data space, then define $\mathcal{H} : \mathcal{X} \mapsto \mathbb{R}^q$ by

$$\mathcal{H}(x(\cdot)) = \mathbb{E}\mathcal{F}(x(\cdot)).$$

We use both moments of the vector x and the averaged auto-correlation function as data:

(i) We will use m^{th}–moments of vector x at time $t = 0$:

$$\mathcal{F}_m(x(\cdot)) = \Pi_{k\in M} x_k(0),$$

where x_k denotes the k^{th} element of vector x, and M is a subset of size m comprising indices (repetition allowed) from $\{1, \cdots, K\}$.

(ii) We will also use autocorrelation function $\mathcal{F}_{ac}(x(\cdot)) = x(t) \otimes x(0)$. In this work, we only consider the autocorrelation of the same element in the vector x, i.e., $x_k(t)x_k(0)$.

6.4.1.2 Single-scale Lorenz 96 Model

We now introduce the single-scale Lorenz 96 system, which does not include the fast variables, to illustrate the combined use of direct and indirect data and the advantage of enforcing conservation constraints in error models. The single-scale Lorenz 96 system describes the evolution of the x_k variables alone,

$$\begin{aligned}\dot{x}_k &= -x_{k-1}(x_{k-2} - x_{k+1}) - x_k + F,\\ x_{k+K} &= x_k,\end{aligned} \tag{6.23}$$

and we use it to generate data in the setting where $K = 36$ and $F = 10$.

As the truncated system, we will only assume that we know the linearized part of the dynamics, resulting in approximate model of the form

$$\begin{aligned} \dot{X}_k &= -X_k + F + \delta(X_{k-2}, X_{k-1}, X_{k+1}, X_{k+2}; \theta_{\mathrm{I}}), \\ X_{k+K} &= X_k. \end{aligned} \tag{6.24}$$

Here, δ is the error model that models the missing quadratic terms; we note that we postulate the need to learn a single universal function δ to account for model error in each component of the equation, reflecting an *a priori* assumption about the homogeneity of the structural error with respect to k. Since the error model δ accounts for the unknown convection term and thus should not introduce additional energy, the state variable X_k is excluded from the inputs of δ in the k^{th} equation. We will use data generated from the untruncated system (6.23) to learn about the error δ in the truncated system (6.24). As data $y = \mathcal{H}(x(t))$ we employ the same types of data (i.e., moments and autocorrelation of the vector x) as described in Section 6.4.2.2.

6.4.2 Numerical Results for Lorenz Models

Before presenting detailed numerical results for Lorenz systems, we summarize several highlights of our numerical results.

1. For a multiscale system with a clear scale separation, local deterministic error model using either direct or indirect data leads to satisfactory model fits. Detailed results are presented in Figs. 6.2 and 6.3.

2. For a multiscale system with less clear scale separation, a local deterministic error model using direct data or indirect data does not lead to a satisfactory model fit. Detailed results are presented in Figs. 6.4 and 6.5. However non-local or stochastic error models do lead to satisfactory fits. Detailed results are presented in Figs. 6.6 to 6.8.

3. For the single-scale Lorenz model we show how an energy constraint can be incorporated into the EKI learning framework; and we show that doing so leads to enhanced calibration of the error model. Detailed results are presented in Figs. 6.9 to 6.11.

6.4.2.1 Lorenz 96 Multi-scale Model

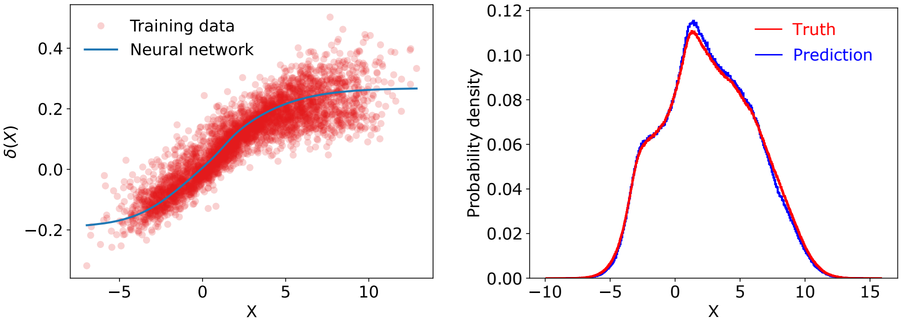
We first studied the multi-scale Lorenz 96 system from (6.19). The numerical examples of multi-scale Lorenz 96 systems are summarized as below:

(i) For $c = 10$ and $h = 1$ in the multi-scale Lorenz 96 system, a dictionary-learning-based model is trained as local deterministic error model $\delta(X_k)$ using direct data ($\{x_k, hc\overline{z}_k\}$), and a fully-connected neural network is trained using indirect data (first and second moments of the slow variable x_k). For the indirect data in this example, we assume partial observation of first eight slow variables and include cross-terms of second moments (i.e., $\mathbb{E}(x_i x_j)$ for different i and j). The results are presented in Figs. 6.2 and 6.3.

(ii) For $c = 3$ and $h = 10/3$ in the multi-scale Lorenz 96 system, the scale separation between fast and slow variables becomes smaller and thus leads to a more challenging case. In this case, a fully-connected neural network is trained as the local deterministic error model $\delta(X_k)$ using either direct data ($\{x_k, hc\overline{z}_k\}$) or indirect data (first to fourth moments of the slow variable x_k and the autocorrelation of x_k). For the indirect data in this and next examples, we enable the full observation of all 36 slow variables and preclude the use of all cross-terms of second to fourth moments. The results are presented in Figs. 6.4 and 6.5.

(iii) For $c = 3$ and $h = 10/3$ in the multi-scale Lorenz 96 system, we trained a non-local deterministic error model $\delta(X) = \sum_{k'} \delta(X_{k'}) C(k - k'; \theta_{\text{non-loc}})$, a local stochastic error model with additive noise $\delta(X_k) + \sqrt{\sigma^2}\dot{W}_k$, and a local stochastic error model with multiplicative noise $\delta(X_k) + \sqrt{\sigma^2(X_k)}\dot{W}_k$, using indirect data (first to fourth moments of the slow variable x_k and the autocorrelation of x_k). The results are presented in Figs. 6.6 to 6.8.

Figure 6.2a presents the direct data $\{x_k, hc\overline{z_k}\}$. Based on direct data, a regression model $\delta(X_k)$ can be trained and then used to simulate the dynamical system of X_k in (6.22). In this work, we train such a regression model using fully-connected neural network with two hidden layers (five neurons at the first hidden layer and one neuron at the second hidden layer). It can be seen in Fig. 6.2a that the trained model captures the general pattern of the training data. We also simulate the dynamical system in (6.22) for a long time trajectory and compare the invariant measure of X_k with the true system in (6.19). As shown in Fig. 6.2b, we obtain a good agreement between the invariant measures of the modeled system and the true system.



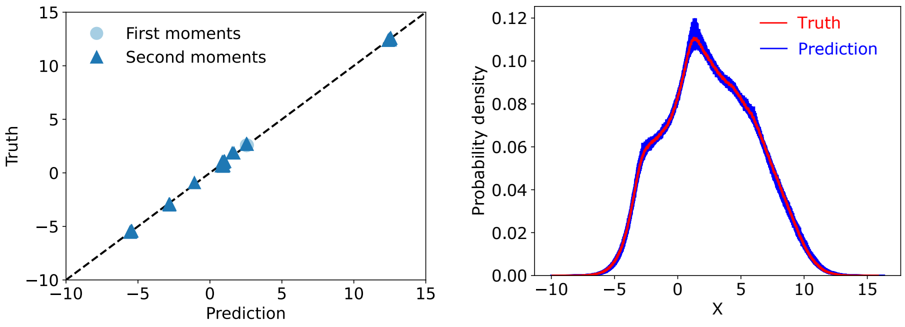
(a) Trained model

(b) Invariant measure

Figure 6.2: Direct training of the error model ($c = 10$) using neural network, with results of (a) the trained error model and (b) invariant measures.

Because direct data $\{x_k, hc\overline{z_k}\}$ may not be accessible for some applications, we also explored the use of indirect data to calibrate the error model $\delta(X_k)$. In this example, the first and second order moments of the first eight components of x_k are used for the calibration. We tested different approaches that parameterize the error model $\delta(X_k)$, including dictionary learning (Fig. 6.3), GPs and neural networks. The error model based on dictionary learning has the form $\delta(X_k) = \sum_{i=1}^{2} \alpha_i \phi_i(X_k)$, where we choose the basis function dictionary $\phi_i(X_k) \in \{\tanh(\beta_1 X_k), \tanh(\beta_2 X_k^2)\}$. Therefore we have $\{\alpha_i, \beta_i\}_{i=1}^{2}$ as unknown parameters that can be learned. Instead of using polynomial basis functions, we have introduced the hyperbolic tangent function $\tanh(\cdot)$ to enhance the numerical stability. The error model based on a GP has the form $\delta(X_k) = \sum_{j=1}^{7} \alpha_j \mathcal{K}(X_k^{(j)}, X_k;\ \psi)$, where we chose the $X_k^{(j)}$ as seven fixed points uniformly distributed in $[-15, 15]$, and $\mathcal{K}$ as a squared exponential kernel with unknown constant hyper-parameters $\psi = \{\sigma_{\text{GP}}, \ell\}$, where σ_{GP} denotes the standard deviation and ℓ the length scale of the kernel. The results with a GP and with a neural network are similar to the ones with dictionary learning in Fig. 6.3 and are omitted here. The calibrated models of all three tests lead to good agreement in both data and invariant measure, and the performance of the calibrated model is not sensitive to the specific choice of parameterization approaches.

Although the performance of the calibrated error model is not sensitive to either the types of data or the parameterization approaches for this numerical example with $c = 10$, we emphasize that the specific choices made in constructing and calibrating error model are still important in general, especially for more challenging scenarios,

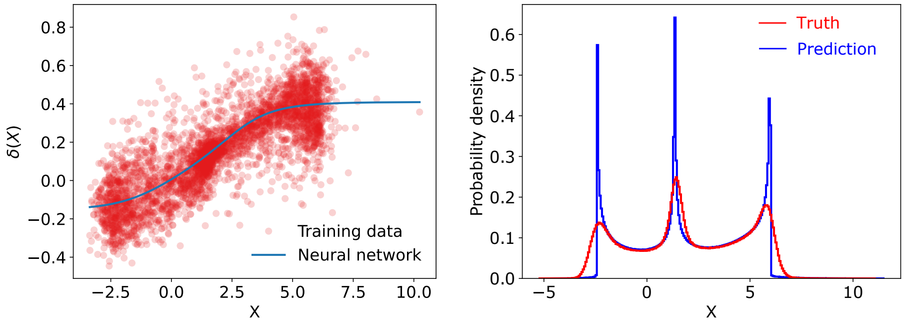


(a) Data comparison

(b) Invariant measure

Figure 6.3: Indirect training of the error model ($c = 10$) using dictionary learning, with the results of (a) first and second order moments and (b) invariant measures. The results with a GP and a neural network have similar performance and are omitted here.

e.g., the resolved and unresolved degrees of freedom have less noticeable scale separation. To illustrate the advantage of using indirect data and stochastic/non-local error model, we studied a more challenging scenario where the scale separation between x_k and $y_{j,k}$ in (6.19) is narrower, by setting $h = 10/3$ and $c = 3$ for both slow and fast dynamics. It can be seen in Fig. 6.4a that the general pattern of direct data is still captured by the trained error model $\delta(X_k)$. However, the comparison of invariant measures in Fig. 6.4b shows that the long-time behaviour of the trained model does not have a good agreement with the true system, indicating the limitation of merely using direct data for the calibration of a local and deterministic error model.

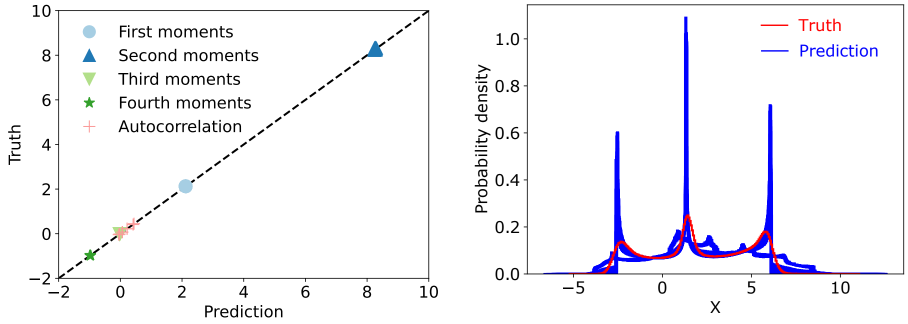


(a) Trained model

(b) Invariant measure

Figure 6.4: Direct training of the error model ($c = 3$) using neural network, with results of (a) the trained error model and (b) invariant measures..

We further investigated the use of indirect data. Specifically, the first four moments of X_k and ten points sampled from the averaged autocorrelation function of X_k are used as training data. Figure 6.5 presents the results of calibrated local model $\delta(X_k)$. It can be seen in Fig. 6.5a that the trained error model provides a good agreement with the training data, while the invariant measures in Fig. 6.5b still demonstrates noticeable difference between the calibrated and the true systems, indicating an overfitting of training data.

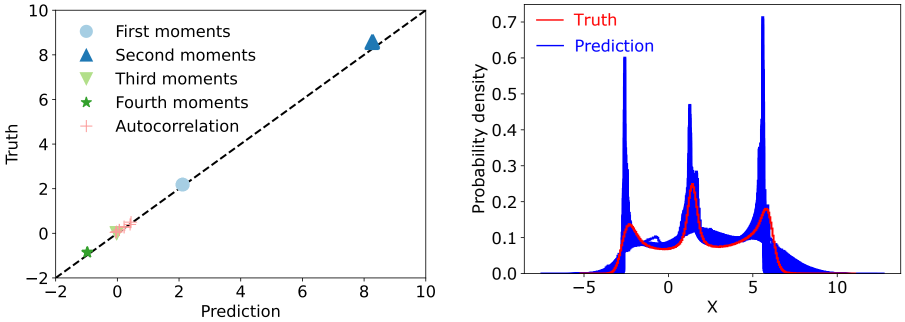


(a) Data comparison (b) Invariant measure

Figure 6.5: Indirect training of the error model ($c = 3$) using deterministic model (local), with the results of (a) first four moments and autocorrelation, and (b) invariant measures.

To avoid the overfitting in Fig. 6.5, we tried to calibrate a non-local error model as discussed in Section 6.3.5. Compared to the results of the local error model, it can be seen in Fig. 6.6 that the invariant measure of the calibrated system has a better agreement with the true system, which indicates that the closure model $\delta(\cdot)$ in (6.22) with non-local effect would better characterize closure for unresolved scales if there is a less clear scale separation between resolved and unresolved scales.

We also explored the learning of stochastic error model for this example. Figure 6.7 presents the results of calibrated system with an additive stochastic error model. Compared to the results of deterministic error models in Figs. 6.5 and 6.6, we can see that the invariant measure of the calibrated system demonstrates a better agreement with the true system in Fig. 6.7b. We further tested the stochastic error model by also learning a state-dependent diffusion coefficient. As shown in Fig. 6.8b, the calibrated system achieves better agreement with the invariant measure of the true system, which confirms that increased flexibility in the stochastic error model can help achieve improved predictive performance via training against indirect data. It should be noted that Figs. 6.6 to 6.8 do not display a single converged invariant measure; this is due to the fact that the calibrated parameters vary across the ensemble, and do

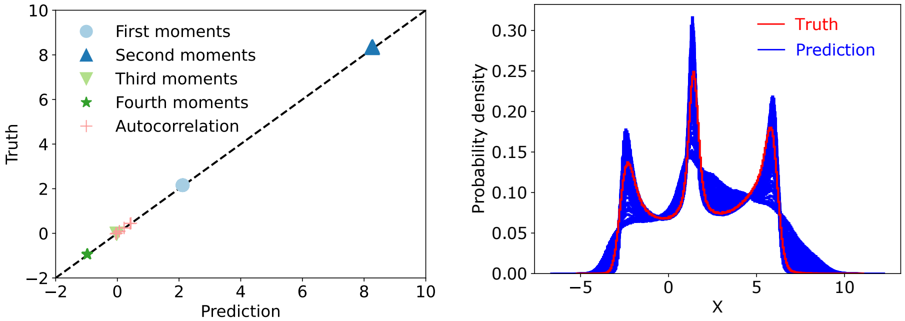


(a) Data comparison

(b) Invariant measure

Figure 6.6: Indirect training of the error model ($c = 3$) using deterministic model (non-local), with the results of (a) first four moments and autocorrelation, and (b) invariant measures.

not reach consensus at a single value; this leads to a family of structural error model that all fit the data with similar accuracy. We surmise that this is caused by the fact that the indirect data contains only partial information about the invariant measure. Nonetheless the fits are all far superior to that obtained with the local deterministic model.



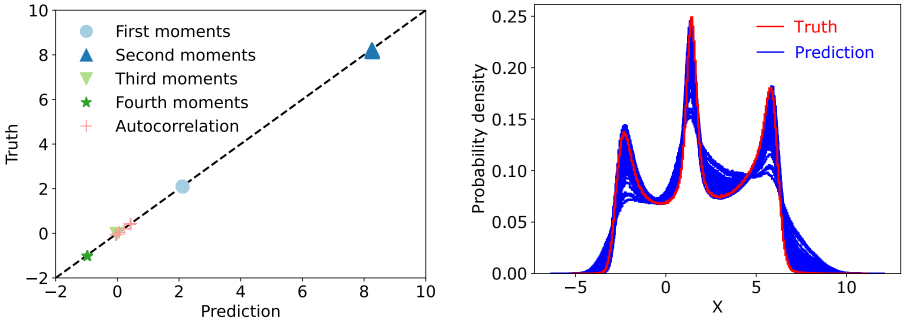
(a) Data comparison

(b) Invariant measure

Figure 6.7: Indirect training of the error model ($c = 3$) using stochastic model with additive noise, with the results of (a) first four moments and autocorrelation, and (b) invariant measures.

6.4.2.2 Lorenz 96 Single-scale Model

We studied the Lorenz 96 single-scale system to learn the quadratic term as an error model. Using this numerical example, we demonstrate the merit of combined use of direct and indirect data and the importance of enforcing physical constraints. More specifically, we assume that the quadratic term of the true system in (6.23) is



(a) Data comparison

(b) Invariant measure

Figure 6.8: Indirect training of the error model ($c = 3$) using stochastic model with multiplicative noise, with the results of (a) first four moments and autocorrelation, and (b) invariant measures.

unknown and then calibrate an error model $\delta(X_{k-2}, X_{k-1}, X_{k+1}, X_{k+2})$ as in (6.24). The numerical examples of the single-scale Lorenz 96 system are summarized as below:

(i) We train a fully-connected neural network as an error model $\delta(\cdot)$ using time-series of x_k and the true quadratic term as direct data. The results are presented in Fig. 6.9.

(ii) We train a fully-connected neural network as an error model $\delta(\cdot)$ using indirect data (first to fourth moments of the state variable x_k and the autocorrelation of x_k). The results are presented in Fig. 6.10.

(iii) We train a fully-connected neural network as an error model $\delta(\cdot)$ using indirect data (first to fourth moments of the state variable x_k and the autocorrelation of x_k) and the energy conservation constraint in (6.12). The results are presented in Fig. 6.11.

Figure 6.9 presents the comparison of invariant measures between the calibrated system and the true system. As we can see in Fig. 6.9, the calibrated system using direct data still demonstrates noticeable differences in the invariant measure, indicating a difference from the long-time behaviour of the true system.

In order to improve the results of Fig. 6.9, we further incorporate indirect data about the x_k. Specifically, we employ the trained model using direct data as the prior mean of EKI, and we set the prior standard deviation as 30% of the mean values for each unknown coefficients of the error model. We then use EKI to calibrate

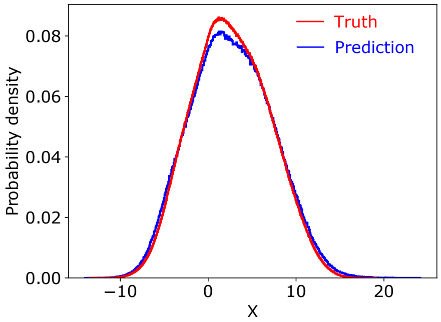
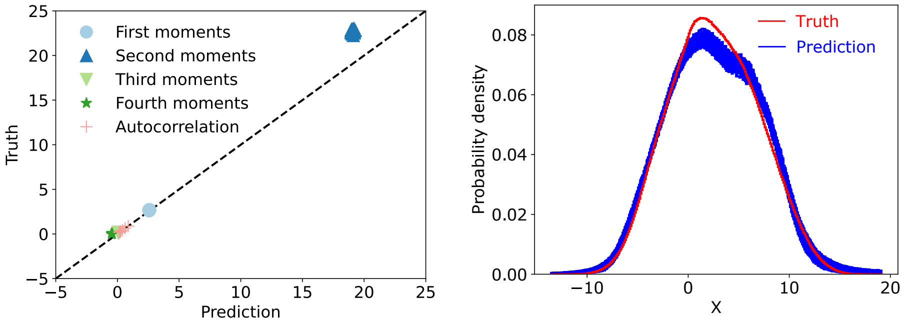


Figure 6.9: Invariant measures via direct training of the error model ($c = 10$) for the single-scale Lorenz 96 system.

the error model based on the first four moments of X_k and the ten sampled points from the autocorrelation function of X_k. Without enforcing energy conservation of the error model, we can see in Fig. 6.10b that the performance of the calibrated model is similar to the calibrated system using direct data in Fig. 6.9. On the other hand, we also performed the calibration based on indirect data and enforced the energy conservation of the error model as discussed in Section 6.2.2.2. As shown in Fig. 6.11, the calibrated error model with energy conservation leads to a modeled system that better fits the training data and achieves a good agreement with the invariant measure of the true system.



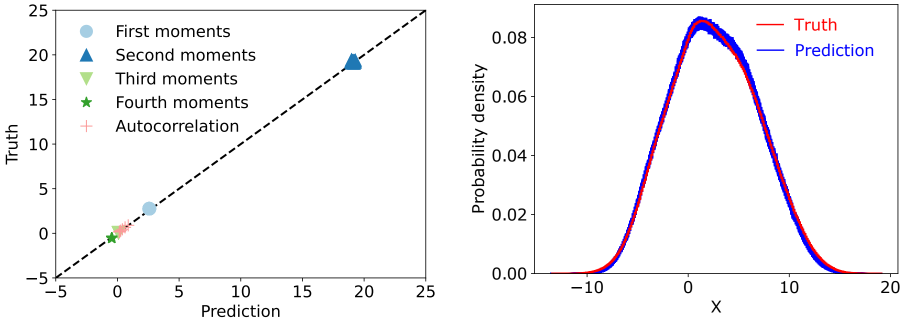
(a) Data comparison

(b) Invariant measure

Figure 6.10: Results from trained error model ($c = 10$) for the single-scale Lorenz 96 system *without* energy conservation constraint, including (a) first four moments and autocorrelation, and (b) invariant measures.

6.5 Human Glucose-Insulin Model as Illustrative Example

We consider the ultradian model of the glucose-insulin system proposed in [398]. Its primary state variables are the plasma glucose concentration, G, the plasma insulin



(a) Data comparison

(b) Invariant measure

Figure 6.11: Results from trained error model ($c = 10$) for the single-scale Lorenz 96 system *with* energy conservation constraint, including (a) first four moments and autocorrelation, and (b) invariant measures.

concentration, I_p, and the interstitial insulin concentration, I_i. We omit the delays proposed in the original work for simplicity. The resulting ordinary differential equations have the form:

$$\frac{dI_p}{dt} = f_1(G) - E\Big(\frac{I_p}{V_p} - \frac{I_i}{V_i}\Big) - \frac{I_p}{t_p} \tag{6.25a}$$

$$\frac{dI_i}{dt} = E\Big(\frac{I_p}{V_p} - \frac{I_i}{V_i}\Big) - \frac{I_i}{t_i} \tag{6.25b}$$

$$\frac{dG}{dt} = f_4(I_p) - f_2(G) - f_3(I_i)G + m_G(t) \tag{6.25c}$$

Here $m_G(t)$ represents a known rate of ingested carbohydrates appearing in the plasma, $f_1(G)$ represents the rate of glucose-dependent insulin production, $f_2(G)$ represents insulin-independent glucose utilization, and $f_3(I_i)G$ represents insulin-dependent glucose utilization. and $f_4(I_p)$ represents insulin-dependent hepatic glucose production. The functional forms of these parameterized processes are

$$f_1(G) = \frac{R_m}{1 + \exp(\frac{-G}{V_g c_1} + a_1)} \text{ : the rate of insulin production} \tag{6.26}$$

$$f_2(G) = U_1\Big(1 - \exp(\frac{-G}{C_2 V_g})\Big) \text{ : insulin-independent glucose utilization} \tag{6.27}$$

$$f_3(I_i) = \frac{1}{C_3 V_g}\Big(U_0 + \frac{U_m - U_0}{1 + (\kappa I_i)^{-\beta}}\Big),\ f_3(I_i)G \text{ : insulin-dependent glucose utilization} \tag{6.28}$$

$$f_4(I_p) = \frac{R_g}{1 + \exp\big(\alpha(\frac{h_3}{C_5 V_p} - 1)\big)} \text{ : insulin-dependent glucose utilization} \tag{6.29}$$

$$\kappa = \frac{1}{C_4}\Big(\frac{1}{V_i} - \frac{1}{E t_i}\Big). \tag{6.30}$$

The functions f_1, f_2, f_3, f_4 are graphed in Figure 6.12, at typical parameter values used in simulations to follow.

The uptake of carbohydrates is modeled by the function

$$m_G(t) = \sum_{j=1}^{N(t)} \frac{m_j k}{60} \exp(k(t_j - t)), \quad N(= \#\{t_j < t\} \tag{6.31}$$

in which N meals occur at times $\{t_j\}_{j=1}^N$, with carbohydrate composition $\{m_j\}_{j=1}^N$. A specifc choice of carbohydrate uptake function, the the resulting model-predicted dynamics of plasma insulin and glucose is shown in in Figure 6.13.

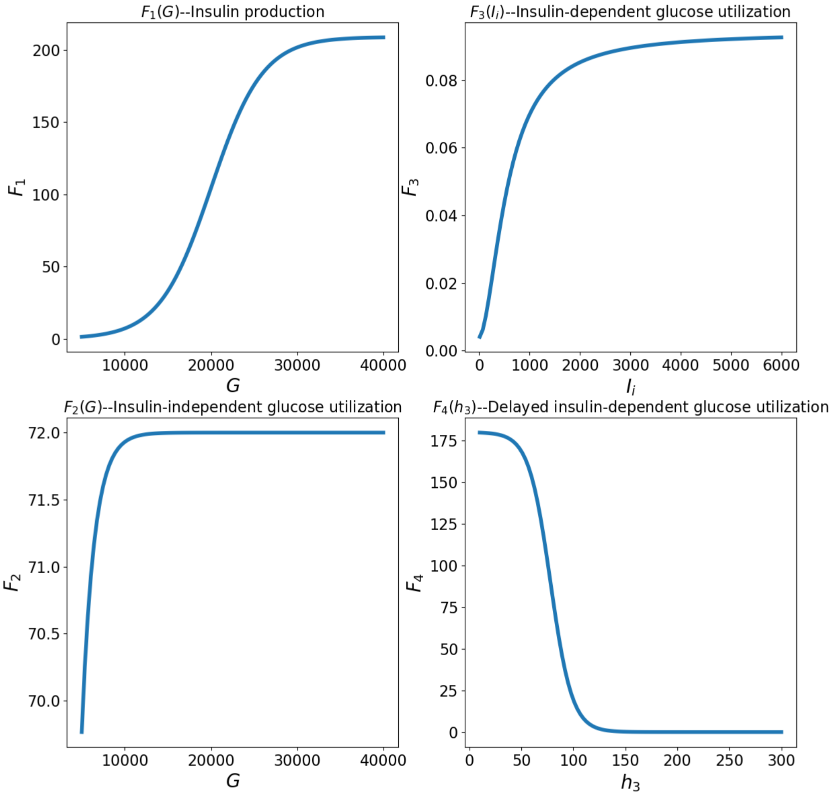


Figure 6.12: Here we show dynamic range for the scalar functions f_1, f_2, f_3, f_4 which appear in (6.25).

Models such as (6.25) are simplifications of complex physiology that is incompletely understood. To improve existing models, we often wish to use data to infer correction terms – that is, to learn about structural model error. Longitudinal clinical data

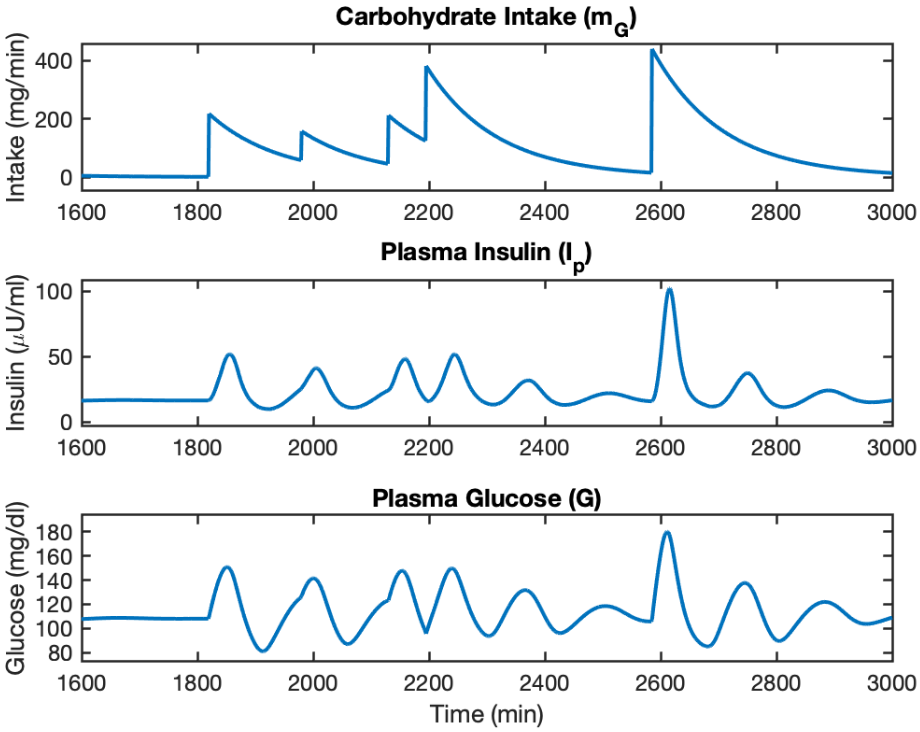


Figure 6.13: Here we show the oscillating dynamics of the glucose-insulin response in the ultradian model, driven by an exponentially decaying nutritional driver m_G.

sets are often available, which include time-series of noisy measurements of blood glucose levels $G(t)$ (sampled at roughly 5 minute intervals) and recorded meal consumption. However, the other modeled states of the patient's physiology are not typically observed: measuring plasma insulin levels, I_p, in the blood is an atypical procedure, although exogenous insulin doses may be known in diabetic sub-populations; and the filter bank variables h_i are analogous to a parameterization in subgrid scale atmospheric models and are not measurable quantities. Because the system is partially and noisily observed, the resulting inference of a structural error model to a system of equations such as (6.25) is indirect in nature. Furthermore we cannot appeal to ergodicity to remove nuisance initialization parameters as we did for the Lorenz '96 model examples considered previously; thus we must recover the unobserved states along with the parameters describing the missing structural model error terms.

We mimic this inference problem in a simulated setting, enabling us to demonstrate a flexible ensemble Kalman based approach to such model error learning. To do this we artificially remove terms from (6.25), resulting in a mis-specified model $\tilde{F}$, then attempt to recover them via data-driven inference. To align with presented notation, let $X = [I_p, I_i, G]$, and use (6.25) to define the true system F such that $\dot{X} = F(X,t)$, where the time-component comes exclusively from $m_G(t)$. For the purposes of this paper, the function $m_G(t)$ may be viewed as known; it is determined from data describing meals consumed by the patient.[4] Measurements are drawn from this system according to

$$Y(t_k) = HX(t_k; X_0) + \eta_k,$$

where $X(t_k; X_0)$ solves the true equations for initialization $X(0) = X_0$, observation operator $H = [0,0,1]$, $\eta_k \sim \mathcal{N}(0, \sigma I)$, and $t_k := hk$ ($h = 5$ minutes) .

We define a mis-specified model by removing f_1 from F, and let $\tilde{F}(x,t) := F(x,t;\ R_m = 0)$. We aim to correct it via an additive structural error model δ

$$\dot{x} = \tilde{F}(x,t) + \delta(x; \theta_I),$$

whose solutions we denote $x(t; \theta_I, x_0, s)$ when initialized at $x(s) = x_0$.

This results in an indirect data inference problem in which we must calibrate a model of form (6.5) from a time-series of noisy, partial observations from the true system in (6.25). Importantly, when calibrating to short-term, partly and noisily observed time-series, it is essential to estimate initial conditions. Concatenating the data to obatin observation vector y and noise to obtain noise vector η we arrive at an inverse problem for (θ_I, x_0) of the form

$$y = \mathcal{G}(\theta_{\mathrm{I}}, x_0) + \eta. \tag{6.32}$$

To allow for a clean exposition we will imagine that the data is given in continuous time (and in practice observing at 5 minute intervals is high frequency on the timescales of the model). This leads to an optimization problem to solve the inverse problem, taking the form

$$J(\theta, x_0) = \frac{1}{T}\int_0^T \|Y(t) - Hx(t; \theta_I, x_0, 0)\|^2 dt. \tag{6.33}$$

To reduce the complexity of this joint-inference problem, it is common to perform alternating descent schemes (Carassi et al. [49]). However, recent methodology

[4] Learning the form of $m_G(\cdot)$ from consumption data is itself an interesting problem[425] but we do not consider it here.

developed in [83] and further explored in [236] suggests that this objective can be well-approximated by a constrained objective that implicitly defines initial condition x_0 as a function of parameters θ and data Y, effectively removing the nuisance parameter x_0, by using data assimilation on parts of the trajectory data:

$$\begin{aligned} \mathcal{J}(\theta_I) &= \frac{1}{T-\tau}\int_{\tau}^{T} \|Y(t) - Hx(t;\theta_I, x_\tau, \tau)\|^2 dt \\ \text{s.t.} \quad x_\tau &= \mathcal{A}\big(\theta_I, \{Y(s)\}_{s=0}^{\tau}\big), \end{aligned} \tag{6.34}$$

where $\mathcal{A}$ is defined by a data assimilation algorithm (e.g., Ensemble Kalman Filter, 3DVAR, etc. [226, 345, 22]) to obtain an estimate x_τ for the filtered state

$$x(\tau) \mid \theta_I, \{Y(s)\}_{s=0}^{\tau}, x(0) = 0.$$

This then defines a $\mathcal{G}$ as

$$\mathcal{G}(\theta_I) := \Big\{Hx\Big(t; \theta_I, \mathcal{A}\big(\theta_I, \{Y(s)\}_{s=0}^{\tau}\big), \tau\Big)\Big\}_{t=\tau}^{T}.$$

Thus, for every evaluation of $\mathcal{J}(\theta_I)$, we first perform data assimilation over an initial window of length τ in order to synchronize our approximate system with the true system. We use the resulting state estimate to initialize a prediction for the remaining data sequence over window of length $T-\tau$, and the value of $\mathcal{J}$ is defined to measure the quality (via path-wise squared error) of this prediction at given parameter θ_I. Ensemble Kalman inversion techniques are well suited to minimizing $\mathcal{J}(\theta_I)$ as they avoid differentiating the complex dependence of $\mathcal{J}(\cdot)$ via the map $\mathcal{A}$. Variants on this approach, for example by estimating initial conditions at a set of points in the interval, may also be used and are discussed in [236]; however the basic form proposed here suffices for our simulation study.

First, we demonstrate in Figure 6.14 that Ensemble Kalman Filtering is a successful data assimilation algorithm when given partial noisy measurements under full, correct knowledge of the governing equations in (6.25). Then we evaluate the quality of state estimation when using mis-specified models. Figures 6.15-6.16 shows the state estimation when removing the f_3 and f_1 term, respectively, from (6.25). We see that the inference is corrupted when removing f_3 (insulin-dependent glucose utilization), but still remains tractable; the removal of f_1, which governs all glucose-dependent production of insulin, is much more impactful, leading to erroneous inferences in the unobserved components.

Next, we will show how we can recover these types of model errors by using EKI to minimize Equation (6.33) [5].

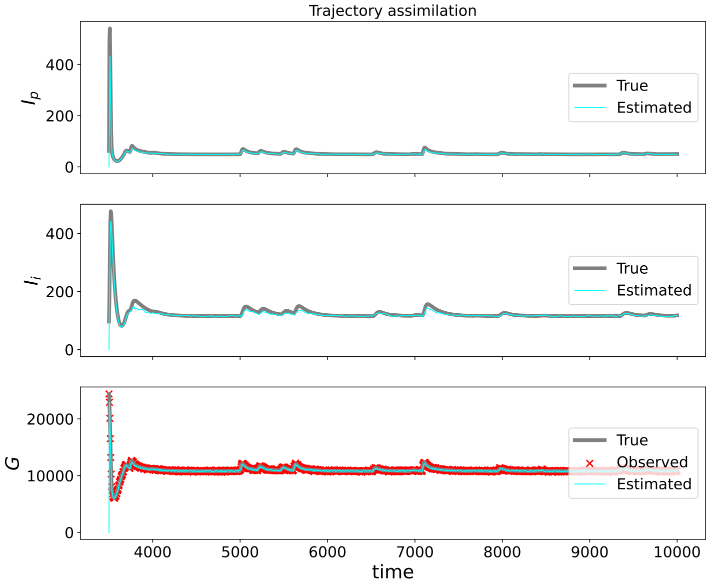


Figure 6.14: Here we show dynamic range for the scalar functions f_1, f_2, f_3, f_4 which appear in (6.25).

6.6 Conclusions

Complex nonlinear dynamics and/or high degrees of freedom present in many complex systems, for example in physiological models of the human body, and turbulent flows around an airplane or in the wake of wind turbines; typically closure models are needed if some degrees of freedom being not resolved by numerical simulations. Without a clear scale separation between resolved and unresolved degrees of freedom, it is expected that most existing models, which are deterministic and local and calibrated by limited amount of data, are not sophisticated enough to capture the true dynamics of the resolved degrees of freedom. Therefore, it is important to study, and learn about, the model error for such complex systems in order to improve the predictive capability of numerical simulations. In this work, we summarize some key aspects of learning model error from data, including the construction of model error and the calibration of model error. In doing so we provide some guidelines about the learning of model error for complex dynamical

[5] Numerical results in preparation and will appear in published version.

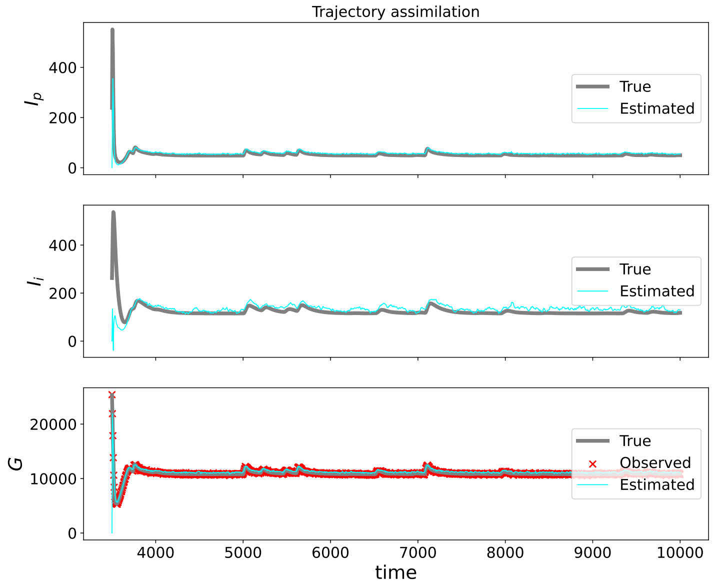


Figure 6.15: Here we show dynamic range for the scalar functions f_1, f_2, f_3, f_4 which appear in (6.25).

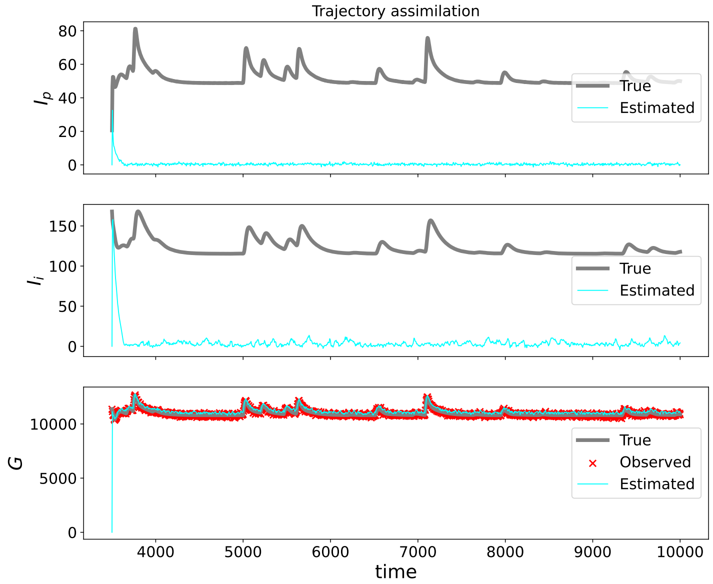


Figure 6.16: Here we show dynamic range for the scalar functions f_1, f_2, f_3, f_4 which appear in (6.25).

systems, ranging from some basic aspects such as different parameterization and calibration techniques (e.g., incorporation of sparsity and physical constraints), to a few advanced aspects such as the combined use of direct and indirect data and the merit of using non-local/stochastic error model. By addressing all these aspects in a systematical manner, our goal is to inspire further applied, methodological and theoretical research in this area; ultimately converge towards a systematic approach of learning model error for complex dynamical systems.

All codes are available at:
https://github.com/jinlong83/Learning-Structural-Errors.git.

6.7 Ensemble Kalman inversion

The use of ensemble Kalman based methods for parameter calibration and the solution of inverse problems, and history of this subject, is overviewed in [Section 4][68]. To be concrete we will concentrate on a particular variant of the methodology, sometimes termed Ensemble Kalman inversion (EKI). This is a specific ensemble-based, gradient-free optimization scheme that was proposed and studied in [188]; we emphasize that other ensemble Kalman based methods share the core desirable attributes of EKI, namely that it is derivative-free, is effective with relatively few evaluations of the forward model $\mathcal{G}$ and is robust to the presence of noise in the evaluations of $\mathcal{G}$.

The core task of EKI is equivalent to a quadratic optimization problem, which facilitates adding linear equality and inequality constraints [4]. To explain the details of EKI, we first introduce a new variable $w = \mathcal{G}(\theta)$ and variables v and $g(v)$:

$$\begin{aligned} v &= (\theta, w)^\top, \\ g(v) &= (\theta, \mathcal{G}(\theta))^\top . \end{aligned} \tag{6.35}$$

Using these variables we formulate the following noisily observed dynamical system:

$$\begin{aligned} v_{m+1} &= g(v_m) \\ y_{m+1} &= Hv_{m+1} + \eta_{m+1}. \end{aligned} \tag{6.36}$$

Here $H = [0, I], H^\perp = [I, 0]$, and hence $Hv = w, H^\perp v = \theta$. In this setting, $\{v_m\}$ is the state and $\{y_m\}$ are the data. The objective is to estimate $H^\perp v_m = \theta_m$ from $\{y_\ell\}_{\ell=1}^m$ and to do so iteratively with respect to m. In practice we only have one data point y and not a sequence y_m; we address this issue in what follows below.

The EKI methodology creates an ensemble $\{v_m^{(j)}\}_{j=1}^J$ defined iteratively in m as follows:

$$
\begin{aligned}
L_m^{(j)}(v) :=& \frac{1}{2}\left|y_{m+1}^{(j)} - Hv\right|_\Gamma^2 + \frac{1}{2}\left|v - g\left(v_m^{(j)}\right)\right|_{C_m^{gg}}^2, \\
v_{m+1}^{(j)} =& \arg\min_v L_m^{(j)}(v).
\end{aligned}
\tag{6.37}
$$

The matrix C^{gg} is the empirical covariance of $\{g(v_m^{(j)})\}_{j=1}^J$. The data $y_{m+1}^{(j)}$ is either fixed so that $y_{m+1}^{(j)} \equiv y$ or created by adding random draws to y from the distribution of the η, independently for all m and j. At each step, m ensemble parameter estimates indexed by $j = 1, \cdots, J$ are found from $\theta_m^{(j)} = H^\perp v_m^{(j)}$.

Using the fact that $v = (\theta, w)^T$, the minimizer $v_{m+1}^{(j)}$ in (6.37) decouples to give the update formula

$$
\theta_{m+1}^{(j)} = \theta_m^{(j)} + C_m^{\theta\mathcal{G}}\left(C_m^{\mathcal{G}\mathcal{G}} + \Gamma\right)^{-1}\left(y_{m+1}^{(j)} - \mathcal{G}(\theta_m^{(j)})\right);
\tag{6.38}
$$

here the matrix $C_m^{\mathcal{G}\mathcal{G}}$ is the empirical covariance of $\{\mathcal{G}(\theta_m^{(j)})\}_{j=1}^J$, while matrix $C_m^{\theta\mathcal{G}}$ is the empirical cross-covariance of $\{\theta_m^{(j)}\}_{j=1}^J$ with $\{\mathcal{G}(\theta_m^{(j)})\}_{j=1}^J$.

To impose sparsity on the solution of θ from EKI, we solve the following constrained optimization problem after each EKI update step:

$$
\begin{aligned}
L_m^{(j)}(v, \lambda) :=& \frac{1}{2}\left|y_{m+1}^{(j)} - Hv\right|_\Gamma^2 + \frac{1}{2}\left|v - g\left(v_m^{(j)}\right)\right|_{C_m^{gg}}^2, \\
v_{m+1}^{(j)} =& \arg\min_{v \in \mathcal{V}} L_m^{(j)}(v),
\end{aligned}
\tag{6.39}
$$

where

$$
\mathcal{V} = \{v : |H^\perp v|_{\ell_1} \leq \gamma\}.
\tag{6.40}
$$

We also employ the thresholding function $\mathcal{T}$ on vectors defined by

$$
\mathcal{T}(\theta_i) = \begin{cases} 0, & \text{if } |\theta_i| < \sqrt{2\lambda} \\ \theta_i, & \text{otherwise,} \end{cases}
\tag{6.41}
$$

to threshold those θ_i with values close to zero, after having solving the constrained optimization problem in (6.39). Such a thresholding step after the ℓ_1-constrained optimization in (6.39) is equivalent to adding ℓ_0 constraint. More details about imposing sparsity into EKI can be found in [374].

Chapter 7

PROTEIN DIMERIZATION NETWORKS AS FUNCTION APPROXIMATORS: EXPRESSIVITY AND ROBUSTNESS

Remark 7.0.1. This chapter is derived from the manuscript in preparation by J Parres-Gold, B Emert, ME Levine, P Perona, AM Stuart, and M Elowitz [189], and contains both excerpts and additions to that work.

7.1 Introduction

In living cells, circuits of interacting proteins compute responses to signals from other cells, the environment, and their own internal state. What functions can these circuits compute? How do they compute them? And why do they use specific circuit architectures to do so? Addressing these questions would both allow the prediction and control of natural cellular behaviors and enable the design of synthetic circuits, such as those engineered to correct disease states [84, 247, 230].

Many natural biochemical circuits employ families of protein variants that interact with one another in a many-to-many fashion. We refer to the binding of two proteins as *dimerization*, which can occur between two *monomers* of the same or different proteins to produce *homo-* and *hetero-dimerized* proteins, respectively. Previous work has suggested that dimerization networks can respond in complex ways to changes in the concentrations of monomer inputs, e.g., [19, 210]. In particular, we focus our attention on responses of dimer concentrations to starting monomer concentrations, as dimerized proteins are typically the biochemically active species in naturally ocurring dimerization networks [16]. We view this input-output response as a *computation* or an execution of a *function* (in the mathematical sense).

However, little is known about the overall range of input-output computations that dimerization networks can perform. Here, we examine the range of input-output functions that can be computed by a simple chemical-equilibrium based model of combinatorial dimerization networks using random parameter screens and targeted optimization trials. These computational experiments allow us to characterize the *expressivity* and *versatility* of input-output maps induced by combinatorial protein dimerization networks, and we study these properties as a function of both network size and connectivity. We use the term *expressivity* to refer to the range of unique input-output functions that a given class of networks can perform. We use the term *versatility* to describe the ability of a single network of proteins to perform different functions in different cell types (which express network components at different abundances).

Figure 7.1 (courtesy of Jacob Parres-Gold) gives a series of representative schematics for the problem setting, which we formalize mathematically in Section 7.2.

7.2 Mathematical setting

Here, we investigate how systems of chemical reactants can transform starting concentrations (e.g., monomeric proteins) into resulting equilibrium concentrations (e.g., dimerized proteins) (see (B) in Figure 7.1).

We begin by introducing a model for chemical reaction kinetics, and prove uniqueness of its equilibrium solution along with its useful re-scaling properties. Then, we specify this model to the case of dimerization networks, and use this formulation to define resulting input-output maps. This allows us to mathematize notions of *expressivity* and *versatility* with respect to the resulting maps.

7.2.1 Chemical reactions: general case

The temporal evolution of concentrations of n chemical components undergoing r simultaneous reactions can be described by the following differential equation:

$$\dot{c}(t) = N^\top v\big(c(t);\ k^+, k^-\big), \quad c(0) = c_0, \tag{7.1}$$

where $c(t) \in \mathbb{R}^n$ denotes the concentration of the n chemical species at time t, $c_0 \in \mathbb{R}^n$ denotes their initial concentrations, $k^+, k^- \in \mathbb{R}^r$ denote the forward and backward equilibrium constants, respectively, for each of the r reactions, $v : \mathbb{R}^n \times \mathbb{R}^r \times \mathbb{R}^r \to \mathbb{R}^r$ computes the instantaneous velocity of the r reactions given a current concentration $c(t)$ and equilibrium constants k^+, k^-, and $N \in \mathbb{R}^{r\times n}$ denotes the stoichiometric matrix associated with the system (which maps the r reaction rates determined by v to rates of change for each of the n individual species). It is a standing assumption that $n \geq r$ and we will in fact assume $n > r$ which is typically the case. Consequently there exists $A \in \mathbb{R}^{(n-r)\times n}$ such that $AN^\top = 0 \in \mathbb{R}^{(n-r)\times r}$. Note that $Ac(t) = Ac_0 \in \mathbb{R}^{n-r}$ because $AN^\top = 0$.

We define v component-wise for each of its r output dimensions as:

$$v_i(c;\ k_i^+, k_i^-) := k_i^+ \prod_{j\in N_i^-} c_j^{|N_{ij}|} - k_i^- \prod_{j\in N_i^+} c_j^{N_{ij}}, \tag{7.2}$$

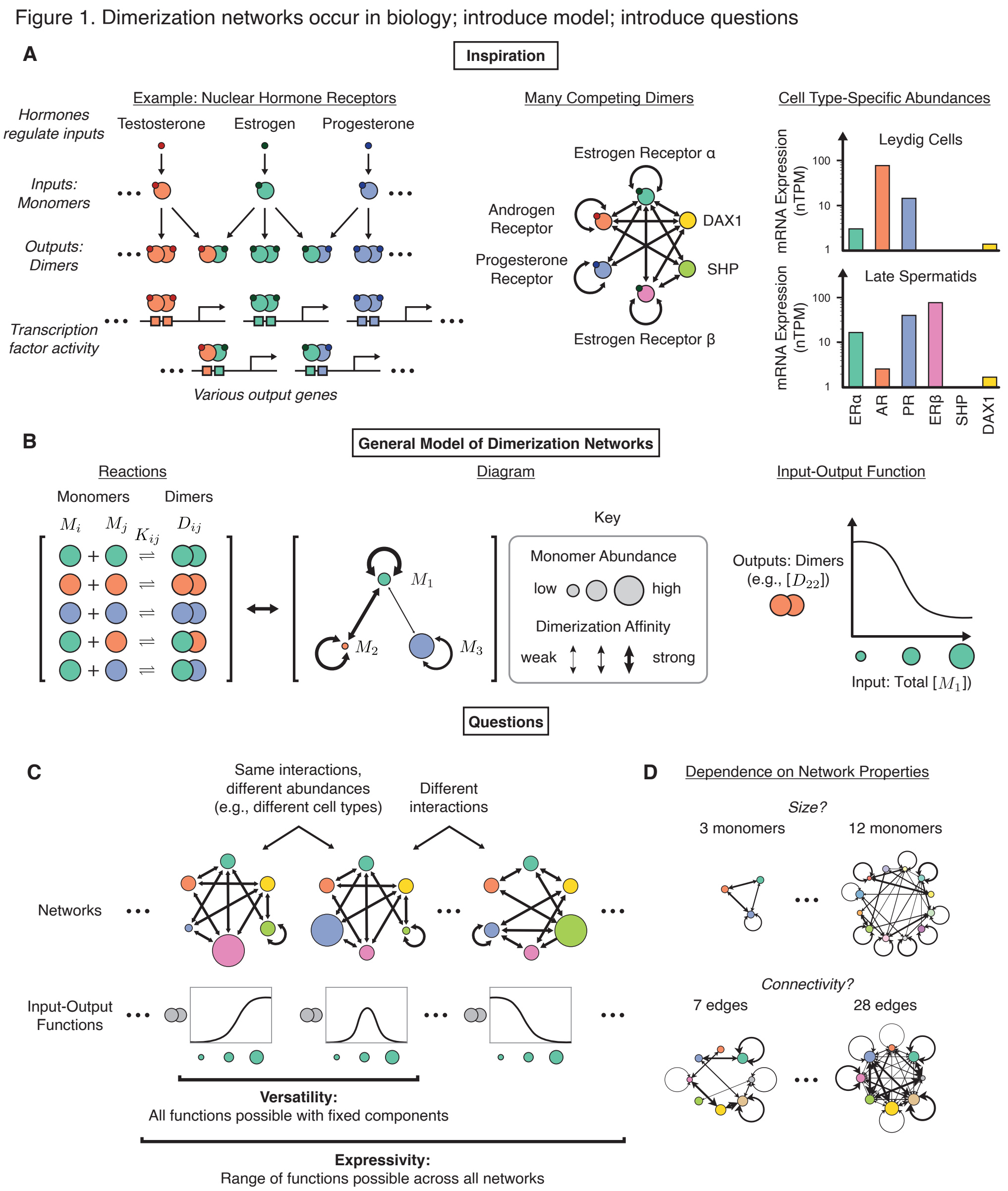


Figure 7.1: (Figure by Jacob Parres-Gold)

where $N_i^- := \{j : N_{ij} < 0\}$, $N_i^+ := \{j : N_{ij} > 0\}$, and $k_i^+, k_i^- \in \mathbb{R}$ denote the forwards and backwards equilibrium rates, respectively, for the i'th reaction.

We can now write the equilibrium equations for Equation (7.1) as

$$\begin{aligned} N^\top v(c;\ k^+, k^-) &= 0, \\ Ac &= Ac_0. \end{aligned} \tag{7.3}$$

This can be simplified to rely only on relative or *effective* equilibrium concentrations. To see this, let $k_i = k_i^+/k_i^-$ and introduce $\tilde{v} : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^r$ component-wise as

$$\tilde{v}_i(c;\ k_i) := k_i \prod_{j \in N_i^-} c_j^{|N_{ij}|} - \prod_{j \in N_i^+} c_j^{N_{ij}}. \tag{7.4}$$

Then, we have

$$v_i(c;\ k_i^+, k_i^-) = k_i^- \tilde{v}_i(c;\ k_i). \tag{7.5}$$

Hence, we simplify our equilibrium equations to be:

$$\begin{aligned} N^\top \tilde{v}(c;\ k) &= 0, \\ Ac &= Ac_0. \end{aligned} \tag{7.6}$$

Theorem 7.2.1. *Fix stochiometric matrix $N \in \mathbb{R}^{r \times n}$, equilibrium constants $k \in \mathbb{R}^r$ and initial conserved quantities $Ac_0 \in \mathbb{R}^{n-r}$. Then there is a unique non-negative solution $c^* \in \mathbb{R}^n$ to Equation (7.6), the equilibrium equations for c.*

See Section 3 of [108] for proof.

Theorem 7.2.2. *Fix stochiometric matrix $N \in \mathbb{R}^{r \times n}$, equilibrium constants $k \in \mathbb{R}^r$ and initial conserved quantities $Ac_0 \in \mathbb{R}^{n-r}$. Let $c^* \in \mathbb{R}^n$ be the unique equilibrium solution to Equation (7.6). Then, for any $\lambda \in \mathbb{R}^+$, λc^* is the unique equilibrium solution to the modified system:*

$$\begin{aligned} N^\top \tilde{v}(c;\ k_\lambda) &= 0, \\ Ac &= \lambda Ac_0, \end{aligned}$$

where

$$k_\lambda := \left[\lambda^{s_1} k_1, \ \cdots, \ \lambda^{s_r} k_r\right],$$

and

$$s_i := \sum_{j=1}^{n} N_{ij}$$

denotes the i'th row sum of stoichiometric matrix N.

Proof. Clearly, the linear constraint is satisfied by taking $c = \lambda c^*$. To verify that λc^* is also an equilibrium, we show that $\tilde{v}_i(\lambda c^*;\ \lambda^{s_i} k_i) = 0$ for all i. We do this by setting (7.4) to 0 and re-arranging terms to yield

$$k_i \lambda^{s_i} \prod_{j \in N_i^-} (\lambda c_j^*)^{|N_{ij}|} = \prod_{j \in N_i^+} (\lambda c_j^*)^{N_{ij}}. \tag{7.7}$$

Take logs of both sides to obtain:

$$\log(k_i) + s_i \log(\lambda) + \sum_{j \in N_i^-} |N_{ij}| \log(\lambda) + \sum_{j \in N_i^-} |N_{ij}| \log(c_j^*) = \sum_{j \in N_i^+} N_{ij} \log(\lambda) + \sum_{j \in N_i^+} N_{ij} \log(c_j^*) \tag{7.8a}$$

$$\log(k_i) + s_i \log(\lambda) - \sum_{j \in N_i^-} N_{ij} \log(\lambda) - \sum_{j \in N_i^-} N_{ij} \log(c_j^*) = \sum_{j \in N_i^+} N_{ij} \log(\lambda) + \sum_{j \in N_i^+} N_{ij} \log(c_j^*) \tag{7.8b}$$

$$\log(k_i) + s_i \log(\lambda) - \sum_{j \in N_i^-} N_{ij} \log(\lambda) - \sum_{j \in N_i^+} N_{ij} \log(\lambda) = \sum_{j \in N_i^+} N_{ij} \log(c_j^*) + \sum_{j \in N_i^-} N_{ij} \log(c_j^*) \tag{7.8c}$$

$$\log(k_i) + \log(\lambda)\left(s_i - \sum_{j=1}^{n} N_{ij}\right) = \sum_{j=1}^{n} N_{ij} \log(c_j^*) \tag{7.8d}$$

By definition of s_i, we have

$$\log(k_i) = \sum_{j=1}^{n} N_{ij} \log(c_j^*) \tag{7.9}$$

and hence

$$\log(k_i) = N_i^\top \log(c^*). \tag{7.10}$$

Observe that $\tilde{v}_i(c^*; k) = 0$ iff (7.10) holds. The original system (i.e. with $\lambda = 1$) satisfies this relation at equilibrium, so the new system also must be at equilibrium. Since λc^* is an equilibrium solution, by Theorem 7.2.1, it is also the unique equilibrium □

7.3 Chemical Reactions: Specific Case

In this section, we specify the general treatment of chemical reaction equilibria in Section 7.2.1 to the context of protein dimerization networks. To start, we let $c^\top = (\mathfrak{m}^\top, \mathfrak{d}^\top) \in \mathbb{R}^n$, with $\mathfrak{m} \in \mathbb{R}^m$ denoting the protein monomers and $\mathfrak{d} \in \mathbb{R}^d$ denoting the protein dimers. Thus $d = \frac{1}{2}m(m+1)$ (we assume no s-mer formation for

$s > 2$, but self-dimerization is allowed). In this setting, to register with the general case in the previous section, we take $n = m + d$ and $r = d$. The d reactions concern only the creation of dimers from monomers, and the reverse reactions also occur, whereby dimers break down into monomers; however it is assumed that there is no direct inter-conversion amongst monomers themselves. If we further assume that the initial concentration of dimers is 0 then we may take the conserved quantities to be $Ac(0) = \mathfrak{m}(0) \in \mathbb{R}^m$ (noting that $m = n - r$ in the notation of the previous section). Furthermore, since the dimers $\mathfrak{d}$ can be viewed as forming the (upper-triangular) part of a symmetric $m \times m$ matrix, we may also rewrite the equilibrium constants vector in terms of a symmetric matrix $k \in \mathbb{R}^{m\times m}_{\text{sym}}$. [1] Finally, we note that specifying this system with a particular m fully defines the stoichiometric matrix N, upto reorderings of indices for monomers and dimers.

Example 7.3.1. Consider monomers A, B and dimers AA, AB and BB. Assume the following chemical reactions:

$$A + A \rightleftharpoons AA$$
$$A + B \rightleftharpoons AB$$
$$B + B \rightleftharpoons BB$$

Thus $m = 2$ and $d = 3$. Let c_1, c_2 denote concentrations of monomers A, B and c_3, c_4, c_5 the concentrations of dimers AA, AB, BB. Assume stochiometric matrix $N \in \mathbb{R}^{3\times 5}$ given by

$$N = \begin{pmatrix} 2 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & -1 & 0 \\ 0 & 2 & 0 & 0 & -1 \end{pmatrix}$$

and chemical reactions which, since they are restricted to creation of dimers from monomers, are doubly indexed:

$$v_{11} = k^+_{11} c_3 - k^-_{11} c_1^2,$$
$$v_{12} = k^+_{12} c_4 - k^-_{12} c_1 c_2,$$
$$v_{22} = k^+_{22} c_5 - k^-_{22} c_2^2.$$

[1] Note that when $\log(\cdot)$ is applied to k to obtain the analog of Theorem 7.2.1, this is done componentwise.

The evolution equations for the chemical concentrations are thus

$$\begin{aligned}
\dot{c}_1 &= 2v_{11} + v_{12}, \quad c_1(0) = c_{1,0}, \\
\dot{c}_2 &= v_{12} + 2v_{22}, \quad c_2(0) = c_{2,0}, \\
\dot{c}_3 &= -v_{11}, \quad c_3(0) = 0, \\
\dot{c}_4 &= -v_{12}, \quad c_4(0) = 0, \\
\dot{c}_5 &= -v_{22}, \quad c_5(0) = 0.
\end{aligned}$$

Note that the conserved quantities (under this dynamics) are

$$\begin{gathered}
c_1 + 2c_3 + c_4, \\
c_2 + c_4 + 2c_5
\end{gathered}$$

and that, since we choose the initial dimer concentrations to be zero, then the conserved quantities are simply equal to $c_{1,0}, c_{2,0}$. The matrix $A \in \mathbb{R}^{2\times 5}$ is thus given by

$$A = \begin{pmatrix} 1 & 0 & 2 & 1 & 0 \\ 0 & 1 & 0 & 1 & 2 \end{pmatrix}$$

Note that $AN^\top = 0 \in \mathbb{R}^{5\times 5}$ as required. The equilibrium equations are

$$\begin{aligned}
k_{11}^{+} c_3 - k_{11}^{-} c_1^2 &= 0 \\
k_{12}^{+} c_4 - k_{12}^{-} c_1 c_2 &= 0 \\
k_{22}^{+} c_5 - k_{22}^{-} c_2^2 &= 0 \\
c_1 + 2c_3 + c_4 &= c_1(0) \\
c_2 + c_4 + 2c_5 &= c_2(0).
\end{aligned}$$

These may be written as a system of five equations parameterized by $k \in \mathbb{R}^{2\times 2}_{\text{sym}}$ with entries k_{11}, k_{12}, k_{22} and by the input values $c_{1,0}, c_{2,0}$ of the conserved quantities. ◊

7.3.1 Defining parametric input-output maps from protein dimerization

We now introduce a function that solves for the equilibrium in Equation (7.6), which we denote $\tilde{g} : \mathbb{R}^n \times \mathbb{R}^r \to \mathbb{R}^n$. Note that $\tilde{g}(c_0;\ k)$ maps an initial species concentration $c_0 \in \mathbb{R}^n$ to the equilibrium value $c^* \in \mathbb{R}^n$ determined by equilibrium constants $k \in \mathbb{R}^d$. These equilibria can be identified numerically (e.g., by fixed point iterations) using publicly available software, `eqtk` [43]; we view this code as an instantiation of $\tilde{g}$.

In the context of protein dimerization networks described in Section 7.3, we have $c_0^\top = \left(\mathfrak{m}^\top(0), \mathfrak{d}^\top(0)\right)$, with $\mathfrak{d}^\top(0) = 0$. To simplify notation, we write the vector of initial conserved quantities as $\mathfrak{m} = \mathfrak{m}(0)$. We further split this vector into $\mathfrak{m}^\top = (x^\top, a^\top)$ where $x \in \mathbb{R}^q$ are referred to as *inputs* and $a \in \mathbb{R}^{m-q}$ are referred to as *accessories*. This defines a new map $g : \mathbb{R}^q \times \mathbb{R}^{m-q} \times \mathbb{R}^d \to \mathbb{R}^d$ via the relation

$$g(x; a, k) := H_d \tilde{g}(c_0, k), \tag{7.12}$$

where $H_d : \mathbb{R}^{d+m} \to \mathbb{R}^d$ is a linear map defined to extract only dimer concentrations (i.e., $\mathfrak{d} = H_d c$) and $c_0^\top = \left(x, a, 0_d\right)$, with 0_d a $d-$length vector of 0's (assuming 0 initial dimer concentrations).

We view $g(x;\ a, k)$ as a function with inputs x that is *parameterized* by both accessory monomer concentrations, a, and binding affinities k. This is motivated by the knowledge that dimerization networks execute a response to a changing situation, communicated by changing concentration levels of protein monomers x. This response is mediated, of course, by the rules governing the reaction; that is, the binding affinities k. More interestingly, it is also thought that other protein monomers a, which we term accessories, are differentially expressed across cell types in order to allow different responses g to the same inputs x. This may enable a single dimerization network to execute fine-tuned, cell-type specific functions.

We further consider scalar-valued output signals given by non-negative linear combinations of dimer outputs:

$$G(x; a, k, \beta) := \beta^T g(x; a, k) \tag{7.13}$$

with $\beta \in \mathbb{R}_+^d$. Throughout this chapter, we will focus on scalar inputs $x \in \mathbb{R}$ such that $q = 1$, but note that higher-dimensional inputs may be very relevant for expressing complex bioligical functions. We will consider scenarios in which β is allowed to take real-values in all entries (i.e., it weights and combines different dimers to produce a single signal), and we will also consider the case when β is all zeros, except for a single entry of a 1 (i.e., it simply selects a particular dimer as an output).

With this perspective, we can ask what sorts of functions $f(x)$ can be expressed by G through varying choices of $\theta = [k, a, \beta] \in \mathbb{R}^{d_\theta}$, with $d_\theta = 2d + (m-1)$. Alternatively, we can specify target functions $f(x)$, and ask what choices of θ best allow us to approximate f.

7.4 Evaluating network expressivity by fitting to a target library

In order to evaluate expressivity of dimerization networks, we define a fixed library $\mathcal{F}$ of 168 target functions (in the form of step functions), and consider the ability of networks of size m to approximate these functions. In this section, we focus on identifying the existence of a pair (k, a) that can produce a dimer response curve similar to a prescribed target function. For each target function f, we define the loss for a given network configuration (k, a) as

$$\mathcal{L}(k, a; f) = \min_{j=1\ldots d} \frac{1}{|D|} \int_D \left| \log\left(f(x)\right) - \log\left(g_j(x; k, a)\right)\right|^2 dx,$$

where the inner minimization over index j allows us to choose a single best dimer across the vector of output functions g. We choose domain $D = [10^{-3}, 10^3] \subset \mathbb{R}$ based on a range of physically relevant input concentrations. We can then define expressivity of size m networks as the average loss across all targets $\mathcal{F} = \{f_1, \ldots, f_N\}$:

$$Q(m; \mathcal{F}) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}\Big(k^*(f_n), a^*(f_n); f_n\Big), \tag{7.14}$$

where

$$k^*(f),\ a^*(f) = \operatorname*{arg\,min}_{k \in \mathcal{K}, a \in \mathcal{A}} \mathcal{L}(k, a; f)$$

are the optimal parameters for approximating a target f. We define constraint set

$$\mathcal{A} := [10^{-3}, 10^3]^{m-1} \subset \mathbb{R}^{m-1}$$

and

$$\mathcal{K} = \{K \in [10^{-7}, 10^5]^d \subset \mathbb{R}^d : K_{22} \geq K_{33} \geq \ldots \geq K_{mm}\}$$

based on realistic physical values. The additional inequality constraints for $\mathcal{K}$ are imposed to remove a permutation invariance of $\mathcal{L}$ with respect to re-ordering of monomer species indices (note that K_{11} is purposely omitted, as this index corresponds to the input monomer, and is thus fixed).

We evaluate $Q(m; \mathcal{F})$ by running independent optimizations for each target $f_n \in \mathcal{F}$ and for each network size m. The estimation of Q requires a sequence of challenging non-convex optimizations over the space of possible (k, a) pairs for each target function and each network size m. To establish an initial lower bound on expressivity, we perform a grid-based search over the parameter space. Then, to refine our estimates, we apply a genetic algorithm initialized in regions of low loss that were identified by the grid-search. We evolve the genetic algorithm to approach a local minimum in the space of (k, a) pairs, and report a refined (more generous) estimate of expressivity at each size m.

Section 7.4 summarizes the results of this experiment, demonstrating that overall expressivity improves with larger network sizes; we see this both in the terms of the mean of all misfits (see Equation (7.14) in (a)) and in terms of the number of library functions that are fit within a specified error tolerance (see (b)). We show that these trends appear when using grid-based optimization, but that our genetic algorithm allows for better fits and stronger estimates of expressivity.

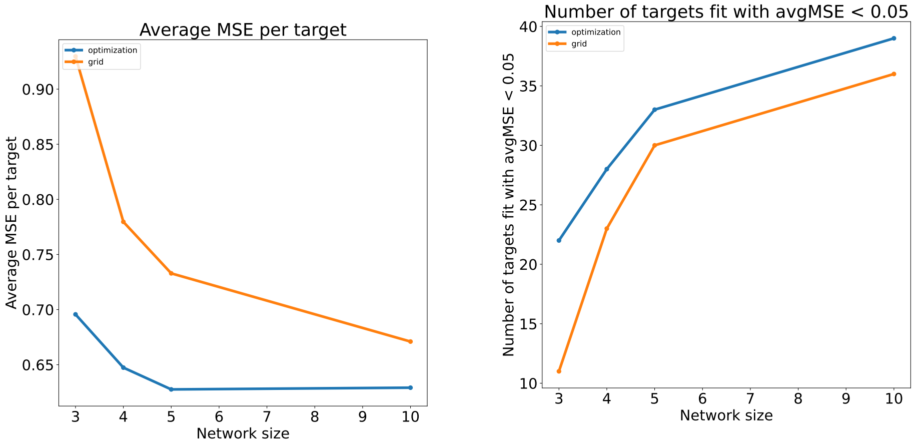
Figure 7.3 focuses on 4 example target functions which were fit using either grid-based (`'x'` marks in the plots) or optimization-based techniques.

- **Target 67:** We find that optimization was able to identify good fits for networks as small as $m = 4$, whereas the grid-based search yielded poorer fits (which started to improve for larger m).

- **Target 110:** This is an example for which all tested network sizes and inference approaches yielded very poor fits. It is likely that this curve is not feasibly expressed by networks within our defined class. Nevertheless, the optimization approach identified better fits than the grid-based methods, especially for $m \in \{4, 5\}$.

- **Target 127:** We find that both optimization and grid-based approaches yield high-fidelity fits to this target function at a variety of network sizes ($m > 3$). For $m = 3$, we identify lower fidelity fits; however, the optimization method yields better fits than a simple grid search.

- **Target 128:** This is an example where both grid-based and optimization-based approaches yield similar fits. Moreover, we observe that fits improve with larger m, and highest fidelity fits are obtained with $m = 10$.



(a) $Q(m)$

(b) Number of target functions fit with MSE < 0.05

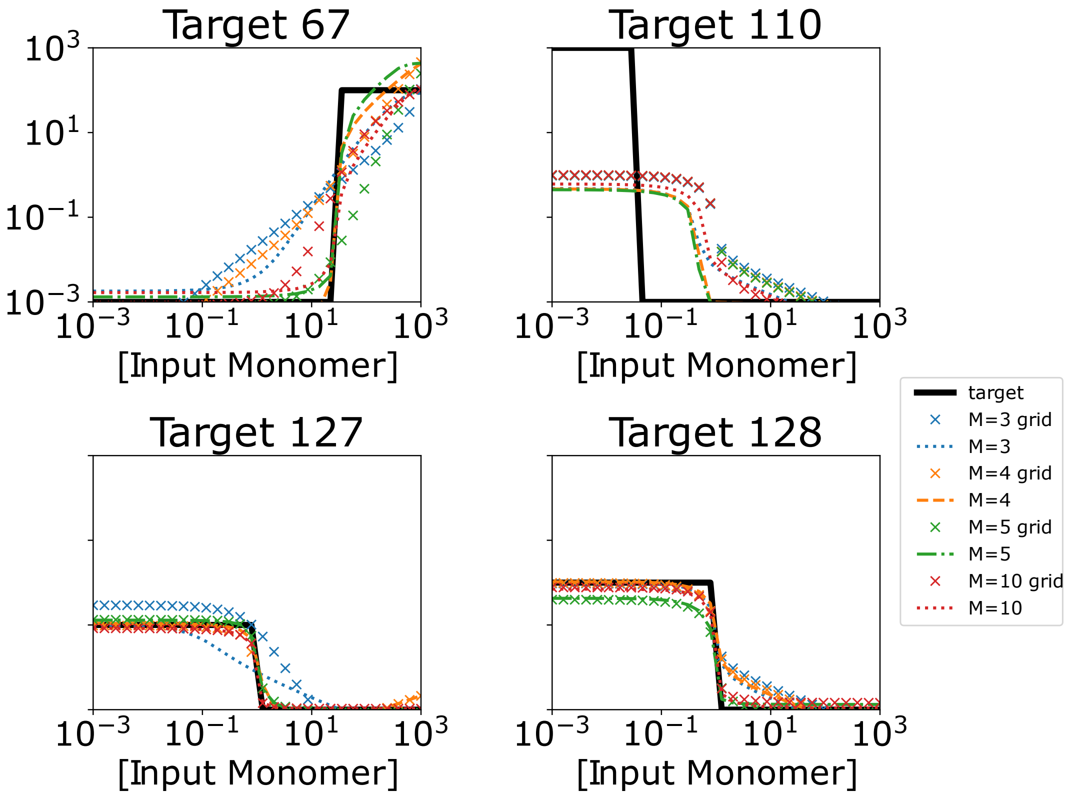


Figure 7.3: short caption

7.4.1 Random Networks: Expressivity and Versatility

In this section, we investigate properties of networks with randomly drawn binding affinities k and tunable accessory concentrations a. We do this by first defining a library of target functions $\mathcal{F}_m$ that were identified by coarsely sampling over (k, a) and clustering the resulting dimer curves into a set of functions $\mathcal{F}_m := \{f_1^{(m)} \dots f_{N_m}^{(m)}\}$, where $N_m := |\mathcal{F}_m|$ [2]. Note that the library differs for each network size m; this allows us to study the fraction of known expressible functions (which may themselves depend on network size m) that can be expressed from a random k and tuned a.

We consider two metrics for evaluating fit quality to a target f using affinity k and dimer index j:

$$\mathcal{L}_2(a;\ j, k, f) = \frac{1}{|D|}\int_D \Big|\log\big(f(x)\big) - \log\big(g_j(x; k, a)\big)\Big|^2 dx, \tag{7.15}$$

and

$$\mathcal{L}_\infty(a;\ j, k, f) = \sup_{x\in D}\Big|\log\big(f(x)\big) - \log\big(g_j(x; k, a)\big)\Big| \tag{7.16}$$

In our experiments, we tune a in order to optimize Equation (7.15), but evaluate the quality of the resulting fit using Equation (7.16); in the future, it is likely best to employ the same metric during both optimization and evaluation. That is, we have

$$a^*(f, j, k) = \arg\min_{a\in\mathcal{A}} \mathcal{L}_2(a;\ j, k, f).$$

We then define a network-size specific *versatility* metric $\mathcal{V}_m$ for a given $k \in \mathbb{R}^d$ (associated with a network of m interacting monomers) and its particular j'th dimer as

$$\mathcal{V}_m(k, j) := \frac{1}{N_m}\sum_{n=1}^{N_m}\left[\mathcal{L}_\infty\Big(a^*\big(f_n^{(m)}, j, k\big);\ j, k, f_n^{(m)}\Big) \le \gamma\right], \tag{7.17}$$

where we choose $\gamma = 1$. Observe that we are summing a boolean, which corresponds to whether or not the j'th dimer resulting from our identified a^* is able to fit the given target within a specified maximum tolerance γ. We term $\mathcal{V}_m$ as versatility because it reports the fraction of target functions that were expressible by a given (k, j) pair; this pair is more versatile if it can express many targets by simply tuning a.

We let $k \sim \mathcal{U}_{\log}\big([10^{-7}, 10^5]^d\big)$, so that $\mathcal{V}_m(k, j)$ becomes a random variable induced by the distribution over k. In order to characterize the distribution of $\mathcal{V}_m$ and its dependence on m, we perform a Monte Carlo sample over k (50 samples), and

[2] Details on the procedure for generating the target library are reported in [189].

subsequently perform the necessary inner optimizations to obtain 50 i.i.d. samples of $\mathcal{V}_m$ (we also evaluate $\mathcal{V}_m$ for every possible j). We visualize the distribution of versatilities in Figure 7.4, which shows that larger random networks possess greater potential for versatility than smaller random networks. Figure 7.5 further shows that this increase in versatility among larger networks is most pronounced when looking at response curves formed by dimerization of the accessory monomers, with heterodimerization the most powerful.

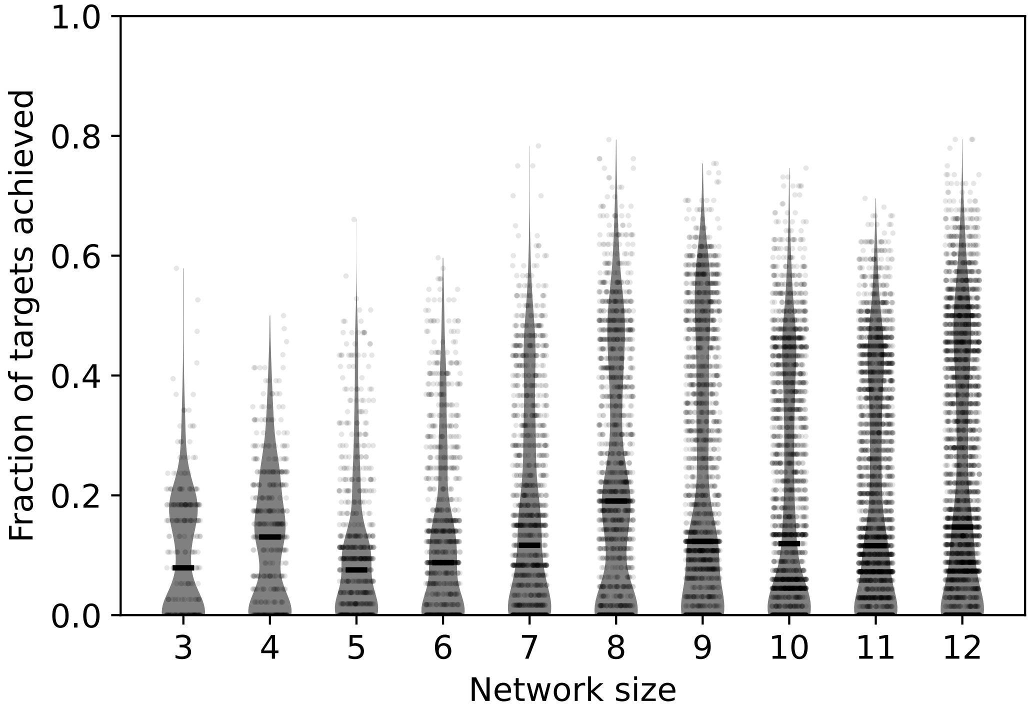


Figure 7.4: Versatility

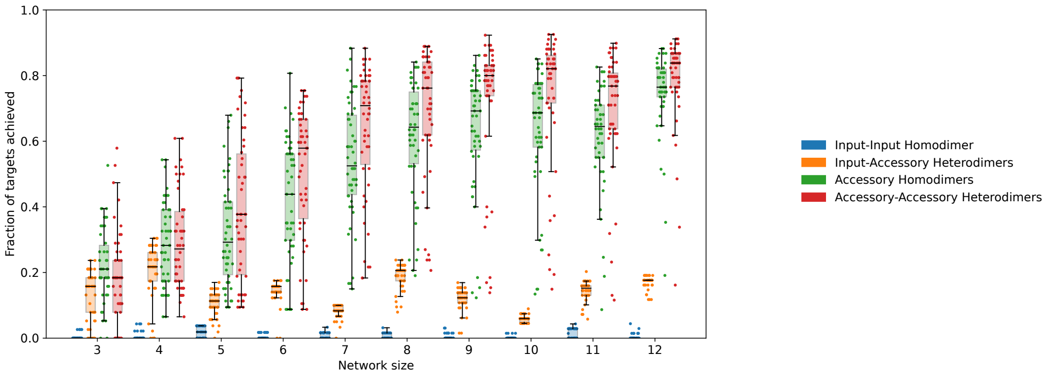


Figure 7.5: Versatility by dimer type

We also study the expressivity of randomly drawn networks by introducing a related experiment. First, we redefine

$$a^*(f,k), j^*(f,k) = \underset{a\in\mathcal{A}, j=1\dots d}{\arg\min}\ \mathcal{L}_2(a;\ j,k,f),$$

which yields an expressivity metric that is independent of dimer index:

$$\mathcal{E}_m(k) := \frac{1}{N_m}\sum_{n=1}^{N_m}\left[\mathcal{L}_\infty\Big(a^*\big(f_n^{(m)},k\big);\ j^*,k,f_n^{(m)}\Big) \le \gamma\right]. \tag{7.18}$$

This expressivity metric allows us to quantify the expressive power of a single network k when given the freedom to use different dimer responses (given by index j) in order to achieve different functions.

Figure 7.6 visualizes the distribution of $\mathcal{E}_m$ induced by 50 randomly drawn networks k. Much like versatility, we observe that expressivity increases in larger networks, and may indeed begin to saturate for $m = 10$.

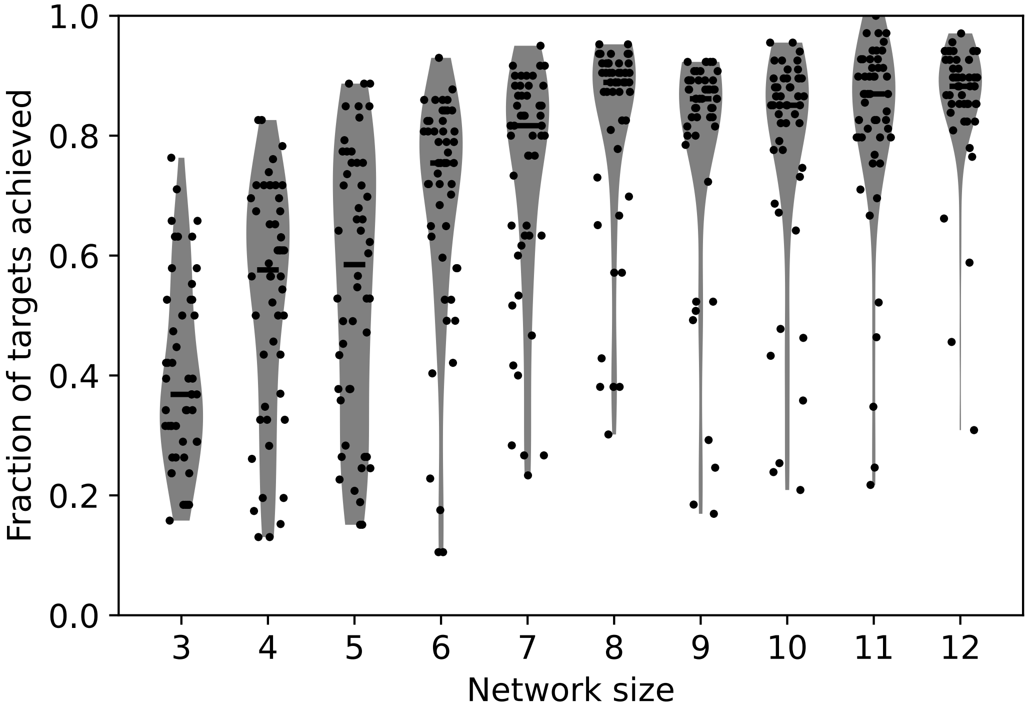


Figure 7.6: Expressivity

BIBLIOGRAPHY

[1] Hasan T Abbas, Lejla Alic, Madhav Erraguntla, Jim X Ji, Muhammad Abdul-Ghani, Qammer H Abbasi, and Marwa K Qaraqe. “Predicting long-term type 2 diabetes with support vector machine using oral glucose tolerance test”. In: *Plos one* 14.12 (2019), e0219636.

[2] Assyr Abdulle, Giacomo Garegnani, Grigorios A Pavliotis, Andrew M Stuart, and Andrea Zanoni. “Drift estimation of multiscale diffusions based on filtered data”. In: *arXiv preprint arXiv:2009.13457* (2020).

[3] D. J. Albers and G. Hripcsak. “Using time-delayed mutual information to discover and interpret temporal correlation structure in complex populations”. In: *CHAOS* 22 (2012), p. 013111.

[4] David J Albers, Paul-Adrien Blancquart, Matthew E Levine, Elnaz Esmaeilzadeh Seylabi, and Andrew M Stuart. “Ensemble Kalman Methods With Constraints”. In: *Inverse Problems* (2019).

[5] David J Albers, Noémie Elhadad, Jan Claassen, Rimma Perotte, A Goldstein, and George Hripcsak. “Estimating summary statistics for electronic health record laboratory data for use in high-throughput phenotyping algorithms”. In: *Journal of biomedical informatics* 78 (2018), pp. 87–101.

[6] David J Albers, Noémie Elhadad, E Tabak, Adler Perotte, and George Hripcsak. “Dynamical phenotyping: using temporal analysis of clinically collected physiologic data to stratify populations”. In: *PloS one* 9.6 (2014), e96443.

[7] David J Albers, George Hripcsak, and Michael Schmidt. “Population physiology: leveraging electronic health record data to understand human endocrine dynamics”. In: *PLoS One* 7.12 (2012), e48058.

[8] David J Albers, Matthew E Levine, Andrew Stuart, Lena Mamykina, Bruce Gluckman, and George Hripcsak. “Mechanistic machine learning: how data assimilation leverages physiologic knowledge using Bayesian inference to forecast the future, infer the present, and phenotype”. In: *Journal of the American Medical Informatics Association* 25.10 (2018), pp. 1392–1401.

[9] David J. Albers, Paul-Adrien Blancquart, Matthew E. Levine, Elnaz Esmaeilzadeh Seylabi, and Andrew Stuart. “Ensemble Kalman methods with constraints”. en. In: *Inverse Problems* 35.9 (Aug. 2019). Publisher: IOP Publishing, p. 095007. ISSN: 0266-5611. DOI: 10.1088/1361-6420/ab1c09. URL: https://arxiv.org/abs/1901.05668 (visited on 04/22/2020).

[10] David J. Albers, Matthew Levine, Bruce Gluckman, Henry Ginsberg, George Hripcsak, and Lena Mamykina. “Personalized Glucose Forecasting for Type 2 Diabetes Using Data Assimilation”. In: *PLOS Computational Biology* 13.4

(Apr. 2017), e1005232. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005232.

[11] David J. Albers, Melike Sirlanci, Matthew E. Levine, Jan Classen, Caroline Der Nigoghossian, and George Hripcsak. “Interpretable Forecasting of Physiology in the ICU Using Constrained Data Assimilation and Electronic Health Record Data”. In: *Journal of Biomedical Informatics (In review)* (2023). arXiv: 2305.06513 [stat.AP].

[12] DJ Albers and G Hripcsak. “Estimation of time-delayed mutual information from sparsely sampled sources”. In: *Chaos, Solitons, and Fractals* 45 (2012), pp. 853–860.

[13] DJ Albers and George Hripcsak. “A statistical dynamics approach to the study of human health data: resolving population scale diurnal variation in laboratory data”. In: *Physics letters A* 374.9 (2010), pp. 1159–1164.

[14] Romeo Alexander and Dimitrios Giannakis. “Operator-theoretic framework for forecasting nonlinear time series with kernel analog techniques”. en. In: *Physica D: Nonlinear Phenomena* 409 (Aug. 2020), p. 132520. ISSN: 0167-2789. DOI: 10.1016/j.physd.2020.132520. URL: http://www.sciencedirect.com/science/article/pii/S016727891930377X (visited on 10/29/2020).

[15] Nesrine Amor, Ghulam Rasool, and Nidhal C Bouaynaya. *Constrained State Estimation — A Review*. 2018. eprint: arXiv:1807.03463.

[16] Grigoris D. Amoutzias, David L. Robertson, Yves Van de Peer, and Stephen G. Oliver. “Choose your partners: dimerization in eukaryotic transcription factors”. In: *Trends in Biochemical Sciences* 33.5 (2008), pp. 220–229. ISSN: 0968-0004. DOI: https://doi.org/10.1016/j.tibs.2008.02.002. URL: https://www.sciencedirect.com/science/article/pii/S0968000408000625.

[17] Ranjan Anantharaman, Yingbo Ma, Shashi Gowda, Chris Laughman, Viral Shah, Alan Edelman, and Chris Rackauckas. “Accelerating Simulation of Stiff Nonlinear Systems using Continuous-Time Echo State Networks”. en. In: (Oct. 2020). URL: https://arxiv.org/abs/2010.04004v6 (visited on 06/08/2021).

[18] Kim E. Andersen and Malene Højbjerre. “A Bayesian Approach to Bergman’s Minimal Model”. In: *International Workshop on Artificial Intelligence and Statistics*. PMLR, Jan. 2003, pp. 1–8.

[19] Yaron E. Antebi, James M. Linton, Heidi Klumpe, Bogdan Bintu, Mengsha Gong, Christina Su, Reed McCardell, and Michael B. Elowitz. “Combinatorial Signal Perception in the BMP Pathway”. In: *Cell* 170.6 (2017), 1184–1196.e24. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2017.08.015. URL: https://www.sciencedirect.com/science/article/pii/S0092867417309406.

[20] HM Arnold, IM Moroz, and TN Palmer. “Stochastic parametrizations and model uncertainty in the Lorenz’96 system”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1991 (2013), p. 20110479.

[21] Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data assimilation: methods, algorithms, and applications*. SIAM, 2016.

[22] Mark Asch, Marc Bocquet, and Maelle Nodet. *Data assimilation: methods, algorithms, and applications*. SIAM, 2016.

[23] Ibrahim Ayed, Emmanuel de Bézenac, Arthur Pajot, Julien Brajard, and Patrick Gallinari. “Learning Dynamical Systems from Partial Observations”. In: *CoRR* abs/1902.11136 (2019).

[24] Ibrahim Ayed, Emmanuel de Bézenac, Arthur Pajot, Julien Brajard, and Patrick Gallinari. *Learning Dynamical Systems from Partial Observations*. Feb. 2019. DOI: 10.48550/arXiv.1902.11136. arXiv: 1902.11136 [physics].

[25] Yunhao Ba, Guangyuan Zhao, and Achuta Kadambi. “Blending Diverse Physical Priors with Neural Networks”. In: *arXiv:1910.00201 [cs, stat]* (Oct. 2019). arXiv: 1910.00201. URL: http://arxiv.org/abs/1910.00201 (visited on 04/05/2021).

[26] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *arXiv:1409.0473 [cs, stat]* (May 2016). arXiv: 1409.0473. URL: http://arxiv.org/abs/1409.0473 (visited on 06/23/2021).

[27] Wael Bahsoun, Ian Melbourne, and Marks Ruziboev. “Variance continuity for Lorenz flows”. In: *Annales Henri Poincare*. Vol. 21. Issue: 6. Springer, 2020, pp. 1873–1892.

[28] Philipp Batz, Andreas Ruttor, and Manfred Opper. “Approximate Bayes learning of stochastic differential equations”. In: *Physical Review E* 98.2 (2018), p. 022109.

[29] Andrea Beck, David Flad, and Claus-Dieter Munz. “Deep neural networks for data-driven LES closure models”. In: *Journal of Computational Physics* 398 (2019), p. 108910.

[30] Randall D. Beer. “On the Dynamics of Small Continuous-Time Recurrent Neural Networks”. en. In: *Adaptive Behavior* 3.4 (Mar. 1995), pp. 469–509. ISSN: 1059-7123, 1741-2633. DOI: 10.1177/105971239500300405. URL: http://journals.sagepub.com/doi/10.1177/105971239500300405 (visited on 02/04/2019).

[31] Pierre-Yves Benhamou, Sylvia Franc, Yves Reznik, Charles Thivolet, Pauline Schaepelynck, Eric Renard, Bruno Guerci, Lucy Chaillous, Celine Lukas-Croisier, Nathalie Jeandidier, et al. "Closed-loop insulin delivery in adults with type 1 diabetes in real-life conditions: a 12-week multicentre, open-label randomised controlled crossover trial". In: *The Lancet Digital Health* 1.1 (2019), e17–e25.

[32] Alain Bensoussan, Jacques-Louis Lions, and George Papanicolaou. *Asymptotic analysis for periodic structures*. Vol. 374. American Mathematical Soc., 2011.

[33] José Bento, Morteza Ibrahimi, and Andrea Montanari. "Information theoretic limits on learning stochastic differential equations". In: *2011 IEEE International Symposium on Information Theory Proceedings*. IEEE, 2011, pp. 855–859.

[34] R N Bergman, Y Z Ider, C R Bowden, and C Cobelli. "Quantitative Estimation of Insulin Sensitivity." In: *American Journal of Physiology-Endocrinology and Metabolism* 236.6 (June 1979), E667. ISSN: 0193-1849, 1522-1555. DOI: 10.1152/ajpendo.1979.236.6.E667.

[35] Richard N Bergman, Y Ziya Ider, Charles R Bowden, and Claudio Cobelli. "Quantitative estimation of insulin sensitivity." In: *American Journal of Physiology-Endocrinology And Metabolism* 236.6 (1979), E667.

[36] Alessandro Bertuzzi, Serenella Salinari, and Geltrude Mingrone. "Insulin granule trafficking in b-cells: mathematical model of glucose-induced insulin secretion". In: *American Journal of Physiology-Endocrinology and Metabolism* (2007).

[37] Tom Beucler, Michael Pritchard, Stephan Rasp, Jordan Ott, Pierre Baldi, and Pierre Gentine. "Enforcing Analytic Constraints in Neural Networks Emulating Physical Systems". In: *Physical Review Letters* 126.9 (2021). Publisher: APS, p. 098302.

[38] Tom Beucler, Michael Pritchard, Stephan Rasp, Jordan Ott, Pierre Baldi, and Pierre Gentine. "Enforcing analytic constraints in neural networks emulating physical systems". In: *Physical Review Letters* 126.9 (2021), p. 098302.

[39] Lucas P Beverlin, Derrick K Rollins, Nisarg Vyas, and David Andre. "An algorithm for optimally fitting a Wiener model". In: *Mathematical problems in Engineering* 2011 (2011).

[40] Kaushik Bhattacharya, Bamdad Hosseini, Nikola B. Kovachki, and Andrew M. Stuart. "Model Reduction and Neural Networks for Parametric PDEs". In: *arXiv:2005.03180 [cs, math, stat]* (May 2020). arXiv: 2005.03180. URL: http://arxiv.org/abs/2005.03180 (visited on 04/15/2021).

[41] Marc Bocquet, Julien Brajard, Alberto Carrassi, and Laurent Bertino. “Bayesian inference of chaotic dynamics by merging data assimilation, machine learning and expectation-maximization”. en. In: *Foundations of Data Science* 2.1 (2020). Company: Foundations of Data Science Distributor: Foundations of Data Science Institution: Foundations of Data Science Label: Foundations of Data Science Publisher: American Institute of Mathematical Sciences, p. 55. DOI: 10.3934/fods.2020004. URL: https://www.aimsciences.org/article/doi/10.3934/fods.2020004 (visited on 04/07/2021).

[42] Marc Bocquet, Julien Brajard, Alberto Carrassi, and Laurent Bertino. “Data assimilation as a learning tool to infer ordinary differential equation representations of dynamical models”. In: *Nonlinear Processes in Geophysics* 26.3 (2019), pp. 143–162.

[43] Justin S. Bois. *justinbois/eqtk: Version 0.1.1*. 2020. DOI: 10.22002/D1.1430. URL: https://data.caltech.edu/records/1430.

[44] Lorenzo Boninsegna, Feliks Nuske, and Cecilia Clementi. “Sparse learning of stochastic dynamical equations”. In: *The Journal of chemical physics* 148.24 (2018), p. 241723.

[45] Vincent Bonnet, Raphaël Dumas, Aurelio Cappozzo, Vladimir Joukov, Gautier Daune, D Kulić, Philippe Fraisse, Sébastien Andary, and Gentiane Venture. “A constrained extended Kalman filter for the optimal estimate of kinematics and kinetics of a sagittal symmetric exercise”. In: *Journal of biomechanics* 62 (2017), pp. 140–147.

[46] Francesco Borra, Angelo Vulpiani, and Massimo Cencini. “Effective models and predictability of chaotic multiscale systems via machine learning”. In: *Physical Review E* 102.5 (Nov. 2020). Publisher: American Physical Society, p. 052203. DOI: 10.1103/PhysRevE.102.052203. URL: https://link.aps.org/doi/10.1103/PhysRevE.102.052203 (visited on 06/02/2021).

[47] Richard J Boys, Darren J Wilkinson, and Thomas BL Kirkwood. “Bayesian inference for a discretely observed stochastic kinetic model”. In: *Statistics and Computing* 18.2 (2008), pp. 125–135.

[48] Julien Brajard, Alberto Carassi, Marc Bocquet, and Laurent Bertino. “Combining Data Assimilation and Machine Learning to Emulate a Dynamical Model from Sparse and Noisy Observations: A Case Study with the Lorenz 96 Model”. In: *Journal of Computational Science* 44 (July 2020), p. 101171. ISSN: 18777503. DOI: 10.1016/j.jocs.2020.101171. arXiv: 2001.01520.

[49] Julien Brajard, Alberto Carassi, Marc Bocquet, and Laurent Bertino. “Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the Lorenz 96 model”. In: *Journal of Computational Science* 44 (July 2020). arXiv: 2001.01520,

p. 101171. ISSN: 18777503. DOI: 10.1016/j.jocs.2020.101171. URL: http://arxiv.org/abs/2001.01520 (visited on 10/30/2020).

[50] Julien Brajard, Alberto Carrassi, Marc Bocquet, and Laurent Bertino. "Combining data assimilation and machine learning to infer unresolved scale parametrization". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2194 (Apr. 2021). Publisher: Royal Society, p. 20200086. DOI: 10.1098/rsta.2020.0086. URL: https://royalsocietypublishing.org/doi/full/10.1098/rsta.2020.0086 (visited on 06/02/2021).

[51] Leo Breiman. "Bagging Predictors". In: *Machine Learning* 24.2 (Aug. 1996), pp. 123–140. ISSN: 1573-0565. DOI: 10.1007/BF00058655.

[52] Troy Bremer and David A Gough. "Is blood glucose predictable from previous values? A solicitation for data." In: *Diabetes* 48.3 (1999), pp. 445–451.

[53] N. D. Brenowitz and C. S. Bretherton. "Prognostic Validation of a Neural Network Unified Physics Parameterization". en. In: *Geophysical Research Letters* 45.12 (2018). _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2018GL078510, pp. 6289–6298. ISSN: 1944-8007. DOI: https://doi.org/10.1029/2018GL078510. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018GL078510 (visited on 04/09/2021).

[54] Noah D Brenowitz, Tom Beucler, Michael Pritchard, and Christopher S Bretherton. "Interpreting and stabilizing machine-learning parametrizations of convection". In: *Journal of the Atmospheric Sciences* 77.12 (2020), pp. 4357–4375.

[55] Noah D Brenowitz and Christopher S Bretherton. "Prognostic validation of a neural network unified physics parameterization". In: *Geophysical Research Letters* 45.12 (2018), pp. 6289–6298.

[56] Christopher S. Bretherton, Brian Henn, Anna Kwa, Noah D. Brenowitz, Oliver Watt-Meyer, Jeremy McGibbon, W. Andre Perkins, Spencer K. Clark, and Lucas Harris. "Correcting Coarse-Grid Weather and Climate Models by Machine Learning From Global Storm-Resolving Simulations". In: *J. Adv. Model. Earth Sys.* 14 (2022), e2021MS002794.

[57] Sue A Brown, Boris P Kovatchev, Dan Raghinaru, John W Lum, Bruce A Buckingham, Yogish C Kudva, Lori M Laffel, Carol J Levy, Jordan E Pinsker, R Paul Wadwa, et al. "Six-month randomized, multicenter trial of closed-loop control in type 1 diabetes". In: *New England Journal of Medicine* 381.18 (2019), pp. 1707–1717.

[58] Paolo Brunetti, Marco Orsini Federici, and Massimo Massi Benedetti. "The Artificial Pancreas". In: *Artificial Cells, Blood Substitutes, and Biotechnology* 31.2 (2003), pp. 127–138. DOI: 10.1081/BIO-120020169. eprint: https://doi.org/10.1081/BIO-120020169. URL: https://doi.org/10.1081/BIO-120020169.

[59] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. “Discovering governing equations from data by sparse identification of nonlinear dynamical systems”. In: *Proceedings of the National Academy of Sciences* 113.15 (2016), pp. 3932–3937.

[60] Steven L. Brunton, Bernd R. Noack, and Petros Koumoutsakos. “Machine Learning for Fluid Mechanics”. In: *Ann. Rev. Fluid Mech.* 52 (2020), pp. 477–508.

[61] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. “Discovering governing equations from data by sparse identification of nonlinear dynamical systems”. en. In: *Proceedings of the National Academy of Sciences* 113.15 (Apr. 2016). Publisher: National Academy of Sciences Section: Physical Sciences, pp. 3932–3937. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1517384113. URL: https://www.pnas.org/content/113/15/3932 (visited on 06/17/2020).

[62] Daniela Bruttomesso. “Toward automated insulin delivery”. In: *N Engl J Med* 381.18 (2019), pp. 1774–1775.

[63] Jenny Brynjarsdottir and Anthony O’Hagan. “Learning about physical parameters: The importance of model discrepancy”. In: *Inverse problems* 30.11 (2014), p. 114007.

[64] Claudia Bucur and Enrico Valdinoci. *Nonlocal diffusion and applications*. Vol. 20. Springer, 2016.

[65] Gerrit Burgers, Peter Jan van Leeuwen, and Geir Evensen. “Analysis scheme in the ensemble Kalman filter”. In: *Monthly weather review* 126.6 (1998), pp. 1719–1724.

[66] Dmitry Burov, Dimitrios Giannakis, Krithika Manohar, and Andrew Stuart. “Kernel Analog Forecasting: Multiscale Test Problems”. In: *arXiv:2005.06623 [physics, stat]* (May 2020). arXiv: 2005.06623. URL: http://arxiv.org/abs/2005.06623 (visited on 11/05/2020).

[67] Jared L Callaham, J-C Loiseau, Georgios Rigas, and Steven L Brunton. “Nonlinear stochastic modelling with Langevin regression”. In: *Proceedings of the Royal Society A* 477.2250 (2021), p. 20210092.

[68] Edoardo Calvello, Sebastian Reich, and Andrew M Stuart. “Ensemble Kalman Methods: A Mean Field Perspective”. In: *arXiv preprint arXiv:2209.11371* (2022).

[69] Daniela Calvetti, Matthew Dunlop, Erkki Somersalo, and Andrew Stuart. “Iterative updating of model error for Bayesian inversion”. In: *Inverse Problems* 34.2 (2018), p. 025008.

[70] Alberto Carrassi, Marc Bocquet, Laurent Bertino, and Geir Evensen. *Data assimilation in the geosciences: An overview of methods, issues, and perspectives*. Vol. 9. 5. Wiley Interdisciplinary Reviews: Climate Change, 5(2018), 2018, e535.

[71] Kathleen Champion, Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. "Data-driven discovery of coordinates and governing equations". en. In: *Proceedings of the National Academy of Sciences* 116.45 (Nov. 2019). Publisher: National Academy of Sciences Section: Physical Sciences, pp. 22445–22451. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1906995116. URL: https://www.pnas.org/content/116/45/22445 (visited on 09/24/2020).

[72] Bo Chang, Minmin Chen, Eldad Haber, and Ed H. Chi. "AntisymmetricRNN: A Dynamical System View on Recurrent Neural Networks". In: *arXiv:1902.09689 [cs, stat]* (Feb. 2019). arXiv: 1902.09689. URL: http://arxiv.org/abs/1902.09689 (visited on 09/04/2020).

[73] Alexis-Tzianni G Charalampopoulos and Themistoklis P Sapsis. "Machine-learning energy-preserving nonlocal closures for turbulent fluid flows and inertial tracers". In: *arXiv preprint arXiv:2102.07639* (2021).

[74] Ashesh Chattopadhyay, Pedram Hassanzadeh, and Devika Subramanian. "Data-driven predictions of a multiscale Lorenz 96 chaotic system using machine-learning methods: reservoir computing, artificial neural network, and long short-term memory network". English. In: *Nonlinear Processes in Geophysics* 27.3 (July 2020). Publisher: Copernicus GmbH, pp. 373–389. ISSN: 1023-5809. DOI: https://doi.org/10.5194/npg-27-373-2020. URL: https://npg.copernicus.org/articles/27/373/2020/ (visited on 07/07/2020).

[75] Ashesh Chattopadhyay, Adam Subel, and Pedram Hassanzadeh. "Data-Driven Super-Parameterization Using Deep Learning: Experimentation With Multiscale Lorenz 96 Systems and Transfer Learning". en. In: *Journal of Advances in Modeling Earth Systems* 12.11 (2020). _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10. e2020MS002084. ISSN: 1942-2466. DOI: https://doi.org/10.1029/2020MS002084. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020MS002084 (visited on 04/12/2021).

[76] Frederick Chee and Tyrone Fernando. *Closed-loop control of blood glucose*. Vol. 368. Springer, 2007.

[77] Nan Chen and Andrew J Majda. "Conditional Gaussian systems for multiscale nonlinear stochastic systems: Prediction, state estimation and uncertainty quantification". In: *Entropy* 20.7 (2018), p. 509.

[78] Nan Chen and Andrew J Majda. "Filtering nonlinear turbulent dynamical systems through conditional Gaussian statistics". In: *Monthly Weather Review* 144.12 (2016), pp. 4885–4917.

[79] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. "Neural Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.

[80] Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M Stuart. *Solving and Learning Nonlinear PDEs with Gaussian Processes*. 2021.

[81] Yifan Chen, Bamdad Hosseini, Houman Owhadi, and Andrew M. Stuart. "Solving and Learning Nonlinear PDEs with Gaussian Processes". In: *arXiv:2103.12959 [cs, math, stat]* (Mar. 2021). arXiv: 2103.12959. URL: http://arxiv.org/abs/2103.12959 (visited on 04/21/2021).

[82] Yuming Chen, Daniel Sanz-Alonso, and Rebecca Willett. "Auto-Differentiable Ensemble Kalman Filters". In: *arXiv:2107.07687 [cs, stat]* (July 2021). arXiv: 2107.07687 [cs, stat].

[83] Yuming Chen, Daniel Sanz-Alonso, and Rebecca Willett. "Autodifferentiable ensemble Kalman filters". In: *SIAM Journal on Mathematics of Data Science* 4.2 (2022), pp. 801–833.

[84] Zibo Chen and Michael B. Elowitz. "Programmable protein circuit design". In: *Cell* 184.9 (2021), pp. 2284–2301. ISSN: 0092-8674. DOI: https://doi.org/10.1016/j.cell.2021.03.007. URL: https://www.sciencedirect.com/science/article/pii/S0092867421002920.

[85] Sai Hung Cheung, Todd A Oliver, Ernesto E Prudencio, Serge Prudhomme, and Robert D Moser. "Bayesian uncertainty analysis with applications to turbulence modeling". In: *Reliability Engineering & System Safety* 96.9 (2011), pp. 1137–1149.

[86] Oksana A Chkrebtii, David A Campbell, Ben Calderhead, Mark A Girolami, et al. "Bayesian solution uncertainty quantification for differential equations". In: *Bayesian Analysis* 11.4 (2016). Publisher: International Society for Bayesian Analysis, pp. 1239–1267.

[87] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches". In: *arXiv:1409.1259 [cs, stat]* (Oct. 2014). arXiv: 1409.1259. URL: http://arxiv.org/abs/1409.1259 (visited on 04/18/2022).

[88] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *arXiv:1406.1078 [cs, stat]* (Sept. 2014). arXiv: 1406.1078. URL: http://arxiv.org/abs/1406.1078 (visited on 06/23/2021).

[89] Alexandre J Chorin, Ole H Hald, and Raz Kupferman. "Optimal prediction and the Mori–Zwanzig representation of irreversible processes". In: *Proceedings of the National Academy of Sciences* 97.7 (2000). Publisher: National Acad Sciences, pp. 2968–2973.

[90] Alexandre J. Chorin and Fei Lu. “Discrete approach to stochastic parametrization and dimension reduction in nonlinear dynamics”. en. In: *Proceedings of the National Academy of Sciences* 112.32 (Aug. 2015). Publisher: National Academy of Sciences Section: Physical Sciences, pp. 9804–9809. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1512080112. URL: https://www.pnas.org/content/112/32/9804 (visited on 08/05/2021).

[91] Gilbert A Churchill Jr. “A paradigm for developing better measures of marketing constructs”. In: *Journal of marketing research* 16.1 (1979), pp. 64–73.

[92] Claudio Cobelli and Alfredo Ruggeri. “Evaluation of portal/peripheral route and of algorithms for insulin delivery in the closed-loop control of glucose in diabetes-A modeling study”. In: *IEEE Transactions on Biomedical Engineering* 2 (1983), pp. 93–103.

[93] David Colton and Rainer Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*. en. 3rd ed. Applied Mathematical Sciences. New York: Springer-Verlag, 2013. ISBN: 978-1-4614-4941-6. DOI: 10.1007/978-1-4614-4942-3. URL: https://www.springer.com/gp/book/9781461449416 (visited on 04/08/2021).

[94] Peter Craven and Grace Wahba. “Smoothing noisy data with spline functions”. In: *Numerische mathematik* 31.4 (1978), pp. 377–403.

[95] G. Cybenko. “Approximation by superpositions of a sigmoidal function”. en. In: *Mathematics of Control, Signals and Systems* 2.4 (Dec. 1989), pp. 303–314. ISSN: 1435-568X. DOI: 10.1007/BF02551274. URL: https://doi.org/10.1007/BF02551274 (visited on 04/14/2021).

[96] Marta D’Elia and Max Gunzburger. “Identification of the diffusion parameter in nonlocal steady diffusion problems”. In: *Applied Mathematics & Optimization* 73.2 (2016), pp. 227–249.

[97] C. Dalla Man, M. Camilleri, and C. Cobelli. “A System Model of Oral Glucose Absorption: Validation on Gold Standard Data”. In: *IEEE Transactions on Biomedical Engineering* 53.12 (Dec. 2006), pp. 2472–2478. ISSN: 0018-9294, 1558-2531. DOI: 10.1109/TBME.2006.883792.

[98] Chiara Dalla Man, Robert A Rizza, and Claudio Cobelli. “Meal simulation model of the glucose-insulin system”. In: *IEEE Transactions on biomedical engineering* 54.10 (2007), pp. 1740–1749.

[99] Chiara Dalla Man, Robert A. Rizza, and Claudio Cobelli. “Meal Simulation Model of the Glucose-Insulin System”. In: *IEEE Transactions on Biomedical Engineering* 54.10 (Oct. 2007), pp. 1740–1749. ISSN: 0018-9294. DOI: 10.1109/TBME.2007.893506.

[100] Matthieu Darcy, Boumediene Hamzi, Jouni Susiluoto, Amy Braverman, and Houman Owhadi. “Learning dynamical systems from data: a simple cross-validation perspective, part II: nonparametric kernel flows”. In: *preprint* (2021).

[101] Eric Darve, Jose Solomon, and Amirali Kia. “Computing generalized Langevin equations and generalized Fokker–Planck equations”. In: *Proceedings of the National Academy of Sciences* 106.27 (2009). Publisher: National Acad Sciences, pp. 10884–10889.

[102] Shaun Davidson, Chris Pretty, Vincent Uyttendaele, Jennifer Knopp, Thomas Desaive, and J Geoffrey Chase. “Multi-input stochastic prediction of insulin sensitivity for tight glycaemic control using insulin sensitivity and blood glucose data”. In: *Computer methods and programs in biomedicine* 182 (2019), p. 105043.

[103] Shaun M Davidson, Vincent Uyttendaele, Christopher G Pretty, Jennifer L Knopp, Thomas Desaive, and J Geoffrey Chase. “Virtual patient trials of a multi-input stochastic model for tight glycaemic control using insulin sensitivity and blood glucose data”. In: *Biomedical Signal Processing and Control* 59 (2020), p. 101896.

[104] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. “Sequential monte carlo samplers”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.3 (2006), pp. 411–436.

[105] Pooja M Desai, Matthew E Levine, David J Albers, and Lena Mamykina. “Pictures worth a thousand words: Reflections on visualizing personal blood glucose forecasts for individuals with type 2 diabetes”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM. 2018, p. 538.

[106] Pooja M Desai, Elliot G Mitchell, Maria L Hwang, Matthew E Levine, David J Albers, and Lena Mamykina. “Personal Health Oracle: Explorations of Personalized Predictions in Diabetes Self-Management”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM. 2019, p. 370.

[107] Ronald A. DeVore and George G. Lorentz. *Constructive Approximation*. en. Grundlehren der mathematischen Wissenschaften. Berlin Heidelberg: Springer-Verlag, 1993. ISBN: 978-3-540-50627-0. URL: https://www.springer.com/gp/book/9783540506270 (visited on 06/02/2021).

[108] Robert M. Dirks, Justin S. Bois, Joseph M. Schaeffer, Erik Winfree, and Niles A. Pierce. “Thermodynamic Analysis of Interacting Nucleic Acid Strands”. In: *SIAM Review* 49.1 (2007), pp. 65–88. DOI: 10.1137/060651100. eprint: https://doi.org/10.1137/060651100. URL: https://doi.org/10.1137/060651100.

[109] Jonathan Dong, Ruben Ohana, Mushegh Rafayelyan, and Florent Krzakala. “Reservoir Computing meets Recurrent Kernels and Structured Transforms”. In: *arXiv:2006.07310 [cs, eess, stat]* (Oct. 2020). arXiv: 2006.07310. URL: http://arxiv.org/abs/2006.07310 (visited on 08/04/2021).

[110] David L Donoho. “Compressed sensing”. In: *IEEE Transactions on information theory* 52.4 (2006), pp. 1289–1306.

[111] J. R. Dormand and P. J. Prince. “A family of embedded Runge-Kutta formulae”. en. In: *Journal of Computational and Applied Mathematics* 6.1 (Mar. 1980), pp. 19–26. ISSN: 0377-0427. DOI: 10.1016/0771-050X(80)90013-3. URL: https://www.sciencedirect.com/science/article/pii/0771050X80900133 (visited on 04/14/2021).

[112] Arnaud Doucet, Nando De Freitas, and Neil Gordon. “An introduction to sequential Monte Carlo methods”. In: *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 3–14.

[113] Qiang Du. *Nonlocal Modeling, Analysis, and Computation: Nonlocal Modeling, Analysis, and Computation*. SIAM, 2019.

[114] Qiang Du, Yiqi Gu, Haizhao Yang, and Chao Zhou. “The Discovery of Dynamics via Linear Multistep Methods and Deep Learning: Error Estimation”. In: *arXiv:2103.11488 [cs, math]* (Mar. 2021). arXiv: 2103.11488. URL: http://arxiv.org/abs/2103.11488 (visited on 07/15/2021).

[115] Qiang Du, Max Gunzburger, R. B. Lehoucq, and Kun Zhou. “Analysis and Approximation of Nonlocal Diffusion Problems with Volume Constraints”. In: *SIAM Review* 54.4 (2012), pp. 667–696.

[116] Oliver RA Dunbar, Andrew B Duncan, Andrew M Stuart, and Marie-Therese Wolfram. “Ensemble inference methods for models with noisy and expensive likelihoods”. In: *SIAM Journal on Applied Dynamical Systems* 21.2 (2022), pp. 1539–1572.

[117] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. “Turbulence modeling in the age of data”. In: *Annual Review of Fluid Mechanics* 51 (2019). Publisher: Annual Reviews, pp. 357–377.

[118] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. “Turbulence modeling in the age of data”. In: *Annual Review of Fluid Mechanics* 51 (2019), pp. 357–377.

[119] Anne Katrine Duun-Henriksen, Signe Schmidt, Rikke Meldgaard Røge, Jonas Bech Møller, Kirsten Nørgaard, John Bagterp Jørgensen, and Henrik Madsen. “Model identification using stochastic differential equation grey-box models in diabetes”. In: *Journal of diabetes science and technology* 7.2 (2013), pp. 431–440.

[120] Weinan E, Chao Ma, and Lei Wu. “A Priori Estimates of the Population Risk for Two-layer Neural Networks”. en. In: *Communications in Mathematical Sciences* 17.5 (2019). arXiv: 1810.06397, pp. 1407–1425. ISSN: 15396746, 19450796. DOI: 10.4310/CMS.2019.v17.n5.a11. URL: http://arxiv.org/abs/1810.06397 (visited on 06/02/2021).

[121] David M Eddy and Leonard Schlessinger. “Archimedes: a trial-validated model of diabetes”. In: *Diabetes care* 26.11 (2003), pp. 3093–3101.

[122] Janet D. Elashoff, Terry J. Reedy, and James H. Meyer. “Analysis of Gastric Emptying Data”. In: *Gastroenterology* 83.6 (Dec. 1982), pp. 1306–1312. ISSN: 00165085. DOI: 10.1016/S0016-5085(82)80145-5.

[123] Michael Emory, Rene Pecnik, and Gianluca Iaccarino. “Modeling structural uncertainties in Reynolds-averaged computations of shock/boundary layer interactions”. In: *49th AIAA Aerospace sciences meeting including the new horizons forum and aerospace exposition*. 2011, p. 479.

[124] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*. Vol. 375. Springer Science & Business Media, 1996.

[125] N. Benjamin Erichson, Omri Azencot, Alejandro Queiruga, Liam Hodgkinson, and Michael W. Mahoney. “Lipschitz Recurrent Neural Networks”. In: *arXiv:2006.12070 [cs, math, stat]* (Oct. 2020). arXiv: 2006.12070. URL: http://arxiv.org/abs/2006.12070 (visited on 10/12/2020).

[126] Geir Evensen. *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media, 2009.

[127] Geir Evensen. “Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics”. In: *Journal of Geophysical Research: Oceans* 99.C5 (1994), pp. 10143–10162.

[128] Geir Evensen. “The ensemble Kalman filter: Theoretical formulation and practical implementation”. In: *Ocean dynamics* 53.4 (2003). Publisher: Springer, pp. 343–367.

[129] Pier Giorgio Fabietti, Valentina Canonico, Marco Orsini Federici, Massimo Massi Benedetti, and Eugenio Sarti. “Control oriented model of insulin and glucose dynamics in type 1 diabetics”. In: *Medical and Biological Engineering and Computing* 44.1-2 (2006), pp. 69–78.

[130] Pier Giorgio Fabietti, Valentina Canonico, Marco Orsini-Federici, Eugenio Sarti, and Massimo Massi-Benedetti. “Clinical validation of a new control-oriented model of insulin and glucose dynamics in subjects with type 1 diabetes”. In: *Diabetes Technology & Therapeutics* 9.4 (2007), pp. 327–338.

[131] Alban Farchi, Patrick Laloyaux, Massimo Bonavita, and Marc Bocquet. “Using machine learning to correct model error in data assimilation and forecast applications”. In: *arXiv:2010.12605 [physics, stat]* (May 2021).

arXiv: 2010.12605. URL: http://arxiv.org/abs/2010.12605 (visited on 06/02/2021).

[132] Ibrahim Fatkullin and Eric Vanden-Eijnden. “A computational strategy for multiscale systems with applications to Lorenz 96 model”. In: *Journal of Computational Physics* 200.2 (2004). Publisher: Elsevier, pp. 605–638.

[133] Ibrahim Fatkullin and Eric Vanden-Eijnden. “A computational strategy for multiscale systems with applications to Lorenz 96 model”. In: *Journal of Computational Physics* 200.2 (2004), pp. 605–638.

[134] Marc Finzi, Ke Alexander Wang, and Andrew G. Wilson. “Simplifying Hamiltonian and Lagrangian Neural Networks via Explicit Constraints”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 13880–13889.

[135] Claude Frankignoul and Klaus Hasselmann. “Stochastic climate models, Part II Application to sea-surface temperature anomalies and thermocline variability”. In: *Tellus* 29.4 (1977), pp. 289–305.

[136] Christian LE Franzke, Terence J O’Kane, Judith Berner, Paul D Williams, and Valerio Lucarini. “Stochastic climate theory and modeling”. In: *Wiley Interdisciplinary Reviews: Climate Change* 6.1 (2015), pp. 63–78.

[137] Brian A. Freno and Kevin T. Carlberg. “Machine-learning error models for approximate solutions to parameterized systems of nonlinear equations”. en. In: *Computer Methods in Applied Mechanics and Engineering* 348 (May 2019), pp. 250–296. ISSN: 0045-7825. DOI: 10.1016/j.cma.2019.01.024. URL: https://www.sciencedirect.com/science/article/pii/S0045782519300490 (visited on 04/01/2021).

[138] Roger Frigola, Yutian Chen, and Carl Edward Rasmussen. “Variational Gaussian Process State-Space Models”. en. In: *Advances in Neural Information Processing Systems* 27 (2014). URL: https://proceedings.neurips.cc/paper/2014/hash/139f0874f2ded2e41b0393c4ac5644f7-Abstract.html (visited on 04/22/2021).

[139] Ken-ichi Funahashi and Yuichi Nakamura. “Approximation of dynamical systems by continuous time recurrent neural networks”. en. In: *Neural Networks* 6.6 (Jan. 1993), pp. 801–806. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(05)80125-X. URL: http://www.sciencedirect.com/science/article/pii/S089360800580125X (visited on 05/25/2020).

[140] Ana Gabriela Gallardo-Hernández, Marcos A. González-Olvera, Medardo Castellanos-Fuentes, Jésica Escobar, Cristina Revilla-Monsalve, Ana Luisa Hernandez-Perez, and Ron Leder. “Minimally-Invasive and Efficient Method to Accurately Fit the Bergman Minimal Model to Diabetes Type 2”. In: *Cellular and Molecular Bioengineering* 15.3 (June 2022), pp. 267–279. ISSN: 1865-5033. DOI: 10.1007/s12195-022-00719-x.

[141] Adiwinata Gani, Andrei V Gribok, Srinivasan Rajaraman, W Kenneth Ward, and Jaques Reifman. “Predicting subcutaneous glucose concentration in humans: data-driven glucose modeling”. In: *IEEE Transactions on Biomedical Engineering* 56.2 (2009), pp. 246–254.

[142] Daniel J. Gauthier, Erik Bollt, Aaron Griffith, and Wendson A. S. Barbosa. “Next Generation Reservoir Computing”. In: *Nature Communications* 12.1 (Sept. 2021), p. 5564. ISSN: 2041-1723. DOI: 10.1038/s41467-021-25801-2.

[143] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

[144] Bryan S Gibson, Sheri R Colberg, Paul Poirier, Denise Maria Martins Vancea, Jason Jones, and Robin Marcus. “Development and validation of a predictive model of acute glucose response to exercise in individuals with type 2 diabetes”. In: *Diabetology & metabolic syndrome* 5.1 (2013), p. 33.

[145] Faheem Gilani, Dimitrios Giannakis, and John Harlim. “Kernel-based prediction of non-Markovian time series”. en. In: *Physica D: Nonlinear Phenomena* 418 (Apr. 2021), p. 132829. ISSN: 0167-2789. DOI: 10.1016/j.physd.2020.132829. URL: https://www.sciencedirect.com/science/article/pii/S0167278920308307 (visited on 06/02/2021).

[146] Pranay Goel. “Insulin resistance or hypersecretion? The βIG picture revisited”. In: *Journal of theoretical biology* 384 (2015), pp. 131–139.

[147] Guillaume Goffaux, Michel Perrier, and Mathieu Cloutier. “Cell energy metabolism: a constrained ensemble Kalman filter”. In: *Proceedings of the 18th IFAC world congress: Milano, Italy, International Federation of Automatic Control*. 2011, pp. 8391–8396.

[148] R. González-García, R. Rico-Martínez, and I.G. Kevrekidis. “Identification of distributed parameter systems: A neural net based approach”. en. In: *Computers & Chemical Engineering* 22 (Mar. 1998), S965–S968. ISSN: 00981354. DOI: 10.1016/S0098-1354(98)00191-4. URL: https://linkinghub.elsevier.com/retrieve/pii/S0098135498001914 (visited on 05/23/2020).

[149] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[150] Georg A. Gottwald and Sebastian Reich. “Combining Machine Learning and Data Assimilation to Forecast Dynamical Systems from Noisy Partial Observations”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.10 (Oct. 2021), p. 101103. ISSN: 1054-1500. DOI: 10.1063/5.0066080.

[151] Georg A. Gottwald and Sebastian Reich. “Supervised learning from noisy observations: Combining machine-learning techniques with data assimilation”. en. In: *Physica D: Nonlinear Phenomena* 423 (Sept. 2021), p. 132911. ISSN:

0167-2789. DOI: 10.1016/j.physd.2021.132911. URL: https://www.sciencedirect.com/science/article/pii/S0167278921000695 (visited on 06/02/2021).

[152] Ayoub Gouasmi, Eric J. Parish, and Karthik Duraisamy. “A priori estimation of memory effects in reduced-order models of nonlinear systems using the Mori–Zwanzig formalism”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473.2205 (Sept. 2017). Publisher: Royal Society, p. 20170385. DOI: 10.1098/rspa.2017.0385. URL: https://royalsocietypublishing.org/doi/10.1098/rspa.2017.0385 (visited on 04/21/2021).

[153] Wojciech W. Grabowski. “Coupling Cloud Processes with the Large-Scale Dynamics Using the Cloud-Resolving Convection Parameterization (CRCP)”. EN. In: *Journal of the Atmospheric Sciences* 58.9 (May 2001). Publisher: American Meteorological Society Section: Journal of the Atmospheric Sciences, pp. 978–997. ISSN: 0022-4928, 1520-0469. DOI: 10.1175/1520-0469(2001)058<0978:CCPWTL>2.0.CO;2. URL: https://journals.ametsoc.org/view/journals/atsc/58/9/1520-0469_2001_058_0978_ccpwtl_2.0.co_2.xml (visited on 04/16/2021).

[154] Lyudmila Grigoryeva and Juan-Pablo Ortega. “Echo state networks are universal”. In: *arXiv:1806.00797 [cs]* (Aug. 2018). arXiv: 1806.00797. URL: http://arxiv.org/abs/1806.00797 (visited on 08/26/2020).

[155] Geoffrey Grimmett and David Stirzaker. *Probability and random processes*. Oxford university press, 2020.

[156] Gerold M Grodsky. “A threshold distribution hypothesis for packet storage of insulin and its mathematical modeling”. In: *The Journal of clinical investigation* 51.8 (1972), pp. 2047–2059.

[157] Abhinav Gupta and Pierre F. J. Lermusiaux. “Neural Closure Models for Dynamical Systems”. In: *arXiv:2012.13869 [physics]* (May 2021). arXiv: 2012.13869. URL: http://arxiv.org/abs/2012.13869 (visited on 06/02/2021).

[158] Joon Ha, Leslie S Satin, and Arthur S Sherman. “A mathematical model of the pathogenesis, prevention, and reversal of type 2 diabetes”. In: *Endocrinology* 157.2 (2015), pp. 624–635.

[159] Joon Ha and Arthur Sherman. “Type 2 Diabetes: One Disease, Many Pathways”. In: *bioRxiv* (2019). DOI: 10.1101/648816. eprint: https://www.biorxiv.org/content/early/2019/11/20/648816.full.pdf. URL: https://www.biorxiv.org/content/early/2019/11/20/648816.

[160] Eldad Haber and Lars Ruthotto. “Stable architectures for deep neural networks”. en. In: *Inverse Problems* 34.1 (Dec. 2017). Publisher: IOP Publishing, p. 014004. ISSN: 0266-5611. DOI: 10.1088/1361-6420/aa9a90.

URL: https://doi.org/10.1088/1361-6420/aa9a90 (visited on 04/06/2021).

[161] Franz Hamilton, Alun L. Lloyd, and Kevin B. Flores. “Hybrid modeling and prediction of dynamical systems”. en. In: *PLOS Computational Biology* 13.7 (July 2017), e1005655. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005655. URL: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005655 (visited on 07/10/2019).

[162] Boumediene Hamzi and Houman Owhadi. “Learning dynamical systems from data: A simple cross-validation perspective, part I: Parametric kernel flows”. en. In: *Physica D: Nonlinear Phenomena* 421 (July 2021), p. 132817. ISSN: 01672789. DOI: 10.1016/j.physd.2020.132817. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167278920308186 (visited on 05/27/2021).

[163] Boumediene Hamzi and Houman Owhadi. “Learning dynamical systems from data: a simple cross-validation perspective, part I: parametric kernel flows”. In: *Physica D: Nonlinear Phenomena* 421 (2021), p. 132817.

[164] David M Hardesty and William O Bearden. “The use of expert judges in scale development: Implications for improving face validity of measures of unobservable constructs”. In: *Journal of Business Research* 57.2 (2004), pp. 98–107.

[165] John Harlim, Shixiao W. Jiang, Senwei Liang, and Haizhao Yang. “Machine learning for prediction with missing dynamics”. en. In: *Journal of Computational Physics* 428 (Mar. 2021), p. 109922. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2020.109922. URL: https://www.sciencedirect.com/science/article/pii/S0021999120306963 (visited on 06/02/2021).

[166] Fabrício P. Härter and Haroldo Fraga de Campos Velho. “Data Assimilation Procedure by Recurrent Neural Network”. In: *Engineering Applications of Computational Fluid Mechanics* 6.2 (Jan. 2012). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/19942060.2012.11015417, pp. 224–233. ISSN: 1994-2060. DOI: 10.1080/19942060.2012.11015417. URL: https://doi.org/10.1080/19942060.2012.11015417 (visited on 06/23/2021).

[167] KLAUS Hasselmann. “PIPs and POPs: The reduction of complex dynamical systems using principal interaction and oscillation patterns”. In: *Journal of Geophysical Research: Atmospheres* 93.D9 (1988), pp. 11015–11021.

[168] Klaus Hasselmann. “Stochastic climate models part I. Theory”. In: *tellus* 28.6 (1976), pp. 473–485.

[169] Niels Haverbeke, Tom Van Herpe, Moritz Diehl, Greet Van den Berghe, and Bart De Moor. “Nonlinear model predictive control with moving horizon state and disturbance estimation-application to the normalization of blood

glucose in the critically ill". In: *IFAC Proceedings Volumes* 41.2 (2008), pp. 9069–9074.

[170] Simon Haykin, Jose C. Principe, Terrence J. Sejnowski, and John Mcwhirter. "Modeling Large Dynamical Systems with Dynamical Consistent Neural Networks". In: *New Directions in Statistical Signal Processing: From Systems to Brains*. Conference Name: New Directions in Statistical Signal Processing: From Systems to Brains. MIT Press, 2007, pp. 203–241. ISBN: 978-0-262-25631-5. URL: https://ieeexplore.ieee.org/document/6282085 (visited on 06/23/2021).

[171] Yanyan He and Dongbin Xiu. "Numerical strategy for model correction using physical constraints". In: *Journal of Computational Physics* 313 (2016), pp. 617–634.

[172] Dan Hendrycks and Kevin Gimpel. "Gaussian Error Linear Units (GELUs)". In: (2016). DOI: 10.48550/ARXIV.1606.08415. URL: https://arxiv.org/abs/1606.08415.

[173] Tom Van Herpe, Marcelo Espinoza, Niels Haverbeke, Bart De Moor, and Greet Van den Berghe. "Glycemia Prediction in Critically Ill Patients Using an Adaptive Modeling Approach". In: *Journal of Diabetes Science and Technology* 1.3 (2007). PMID: 19885089, pp. 348–356. DOI: 10.1177/193229680700100306. URL: https://doi.org/10.1177/193229680700100306.

[174] Pau Herrero, Jorge Bondia, Cesar C. Palerm, Josep Vehí, Pantelis Georgiou, Nick Oliver, and Christofer Toumazou. "A Simple Robust Method for Estimating the Glucose Rate of Appearance from Mixed Meals". In: *Journal of Diabetes Science and Technology* 6.1 (Jan. 2012), pp. 153–162. ISSN: 1932-2968, 1932-2968. DOI: 10.1177/193229681200600119.

[175] Dave Higdon, Marc Kennedy, James C Cavendish, John A Cafeo, and Robert D Ryne. "Combining field data and computer simulations for calibration and prediction". In: *SIAM Journal on Scientific Computing* 26.2 (2004), pp. 448–466.

[176] Carmen Hijón, Pep Español, Eric Vanden-Eijnden, and Rafael Delgado-Buscalioni. "Mori–Zwanzig formalism as a practical computational tool". In: *Faraday discussions* 144 (2010). Publisher: Royal Society of Chemistry, pp. 301–322.

[177] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[178] Mark Holland and Ian Melbourne. "Central limit theorems and invariance principles for Lorenz attractors". In: *Journal of the London Mathematical Society* 76.2 (2007). Publisher: Oxford University Press, pp. 345–364.

[179] AAM Holtslag and Chin-Hoh Moeng. “Eddy diffusivity and countergradient transport in the convective atmospheric boundary layer”. In: *Journal of the Atmospheric Sciences* 48.14 (1991), pp. 1690–1698.

[180] Christoph Honisch and Rudolf Friedrich. “Estimation of Kramers-Moyal coefficients at low sampling rates”. In: *Physical Review E* 83.6 (2011), p. 066701.

[181] Roman Hovorka, Ludovic J Chassin, Martin Ellmerer, Johannes Plank, and Malgorzata E Wilinska. “A simulation model of glucose regulation in the critically ill”. In: *Physiological measurement* 29.8 (2008), p. 959.

[182] Roman Hovorka et al. “Nonlinear Model Predictive Control of Glucose Concentration in Subjects with Type 1 Diabetes”. In: *Physiological Measurement* 25.4 (Aug. 2004), pp. 905–920. ISSN: 0967-3334, 1361-6579. DOI: 10.1088/0967-3334/25/4/010.

[183] George Hripcsak and David J Albers. “Correlating electronic health record concepts with healthcare process events”. In: *Journal of the American Medical Informatics Association* 20.e2 (2013), e311–e318.

[184] George Hripcsak and David J Albers. “High-fidelity phenotyping: richness and freedom from bias”. In: *Journal of the American Medical Informatics Association* 25.3 (2017), pp. 289–294.

[185] George Hripcsak and David J Albers. “Next-generation phenotyping of electronic health records”. In: *Journal of the American Medical Informatics Association* 20.1 (2012), pp. 117–121.

[186] George Hripcsak, David J Albers, and Adler Perotte. “Exploiting time in electronic health record correlations”. In: *Journal of the American Medical Informatics Association* 18.Supplement_1 (2011), pp. i109–i115.

[187] D Huang, ME Levine, T Schneider, Z Shen, AM Stuart, and J Wu. “Learning About Structural Errors in Models of Complex Dynamical Systems”. In: *In preparation* (2023).

[188] Marco A Iglesias, Kody JH Law, and Andrew M Stuart. “Ensemble Kalman methods for inverse problems”. In: *Inverse Problems* 29.4 (2013), p. 045001.

[189] J Parres-Gold, B Emert, ME Levine, P Perona, AM Stuart, and M Elowitz. “On the expressivity of combinatorial protein dimerization networks”. In: *In preparation* (2023).

[190] Herbert Jaeger. “The" echo state" approach to analysing and training recurrent neural networks-with an erratum note’”. In: *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148 (Jan. 2001).

[191] Tijana Janjić, Dennis McLaughlin, Stephen E Cohn, and Martin Verlaan. "Conservation of mass and preservation of positivity with ensemble-type Kalman filter algorithms". In: *Monthly Weather Review* 142.2 (2014), pp. 755–773.

[192] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan S. Read, Jacob A. Zwart, Michael Steinbach, and Vipin Kumar. "Physics-Guided Machine Learning for Scientific Discovery: An Application in Simulating Lake Temperature Profiles". In: *ACM/IMS Transactions on Data Science* 2.3 (May 2021), 20:1–20:26. ISSN: 2691-1922. DOI: 10.1145/3447814. URL: https://doi.org/10.1145/3447814 (visited on 06/02/2021).

[193] Shixiao W. Jiang and John Harlim. "Modeling of Missing Dynamical Systems: Deriving Parametric Models using a Nonparametric Framework". In: *Research in the Mathematical Sciences* 7.3 (Sept. 2020). arXiv: 1905.08082, p. 16. ISSN: 2522-0144, 2197-9847. DOI: 10.1007/s40687-020-00217-4. URL: http://arxiv.org/abs/1905.08082 (visited on 09/28/2020).

[194] Simon Julier, Jeffrey Uhlmann, and Hugh F Durrant-Whyte. "A new method for the nonlinear transformation of means and covariances in filters and estimators". In: *IEEE Transactions on automatic control* 45.3 (2000). Publisher: IEEE, pp. 477–482.

[195] Kadierdan Kaheman, Steven L. Brunton, and J. Nathan Kutz. "Automatic differentiation to simultaneously identify nonlinear dynamics and extract noise probability distributions from data". en. In: *Machine Learning: Science and Technology* 3.1 (Mar. 2022). Publisher: IOP Publishing, p. 015031. ISSN: 2632-2153. DOI: 10.1088/2632-2153/ac567a. URL: https://doi.org/10.1088/2632-2153/ac567a (visited on 03/08/2022).

[196] Kadierdan Kaheman, Eurika Kaiser, Benjamin Strom, J. Nathan Kutz, and Steven L. Brunton. "Learning Discrepancy Models From Experimental Data". en. In: *arXiv:1909.08574 [cs, eess, stat]* (Sept. 2019). arXiv: 1909.08574. URL: http://arxiv.org/abs/1909.08574 (visited on 11/25/2019).

[197] Steven E Kahn, Ronald L Prigeon, David K McCulloch, Edward J Boyko, Richard N Bergman, Michael W Schwartz, James L Neifing, W Kenneth Ward, James C Beard, and Jerry P Palmer. "The contribution of insulin-dependent and insulin-independent glucose uptake to intravenous glucose tolerance in healthy human subjects". In: *Diabetes* 43.4 (1994), pp. 587–592.

[198] Jari Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. en. Applied Mathematical Sciences. New York: Springer-Verlag, 2005. ISBN: 978-0-387-22073-4. DOI: 10.1007/b138659. URL: https://www.springer.com/gp/book/9780387220734 (visited on 06/15/2021).

[199] Jari Kaipio and Erkki Somersalo. *Statistical and computational inverse problems*. Vol. 160. Springer Science & Business Media, 2006.

[200] Serafim Kalliadasis, Sebastian Krumscheid, and Grigorios A Pavliotis. “A new framework for extracting coarse-grained models from time series with multiscale structure”. In: *Journal of Computational Physics* 296 (2015), pp. 314–328.

[201] Rudolph Emil Kalman et al. “A new approach to linear filtering and prediction problems”. In: *Journal of basic Engineering* 82.1 (1960), pp. 35–45.

[202] J. Nagoor Kani and Ahmed H. Elsheikh. “DR-RNN: A deep residual recurrent neural network for model reduction”. In: *arXiv:1709.00939 [cs]* (Sept. 2017). arXiv: 1709.00939. URL: http://arxiv.org/abs/1709.00939 (visited on 09/12/2020).

[203] K. Kashinath et al. “Physics-informed machine learning: case studies for weather and climate modelling”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2194 (Apr. 2021). Publisher: Royal Society, p. 20200093. DOI: 10.1098/rsta.2020.0093. URL: https://royalsocietypublishing.org/doi/full/10.1098/rsta.2020.0093 (visited on 02/26/2021).

[204] Rachael T. Keller and Qiang Du. “Discovery of Dynamics Using Linear Multistep Methods”. In: *SIAM Journal on Numerical Analysis* 59.1 (Jan. 2021). Publisher: Society for Industrial and Applied Mathematics, pp. 429–455. ISSN: 0036-1429. DOI: 10.1137/19M130981X. URL: https://epubs.siam.org/doi/10.1137/19M130981X (visited on 07/15/2021).

[205] Felix P. Kemeth, Tom Bertalan, Nikolaos Evangelou, Tianqi Cui, Saurabh Malani, and Ioannis G. Kevrekidis. “Initializing LSTM Internal States via Manifold Learning”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 31.9 (Sept. 2021), p. 093111. ISSN: 1054-1500, 1089-7682. DOI: 10.1063/5.0055371. arXiv: 2104.13101.

[206] Marc C Kennedy and Anthony O’Hagan. “Bayesian calibration of computer models”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.3 (2001), pp. 425–464.

[207] Marat F. Khairoutdinov and David A. Randall. “A cloud resolving model as a cloud parameterization in the NCAR Community Climate System Model: Preliminary results”. en. In: *Geophysical Research Letters* 28.18 (2001). _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2001GL013552, pp. 3617–3620. ISSN: 1944-8007. DOI: https://doi.org/10.1029/2001GL013552. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2001GL013552 (visited on 04/16/2021).

[208] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[209] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015. URL: http://arxiv.org/abs/1412.6980.

[210] Heidi E. Klumpe, Matthew A. Langley, James M. Linton, Christina J. Su, Yaron E. Antebi, and Michael B. Elowitz. "The context-dependent, combinatorial logic of BMP signaling". In: *Cell Systems* 13.5 (2022), 388–407.e10. ISSN: 2405-4712. DOI: https://doi.org/10.1016/j.cels.2022.03.002. URL: https://www.sciencedirect.com/science/article/pii/S2405471222001296.

[211] Timothy D Knab, Gilles Clermont, and Robert S Parker. "A "Virtual Patient" Cohort and Mathematical Model of Glucose Dynamics in Critical Care". In: *IFAC-PapersOnLine* 49.26 (2016), pp. 1–7.

[212] Timothy D Knab, Gilles Clermont, and Robert S Parker. "Zone model predictive control and moving horizon estimation for the regulation of blood glucose in critical care patients". In: *IFAC-PapersOnLine* 48.8 (2015), pp. 1002–1007.

[213] Juš Kocijan. *Modelling and Control of Dynamic Systems Using Gaussian Process Models*. en. Advances in Industrial Control. Springer International Publishing, 2016. ISBN: 978-3-319-21020-9. DOI: 10.1007/978-3-319-21021-6. URL: https://www.springer.com/gp/book/9783319210209 (visited on 04/22/2021).

[214] Milan Korda, Mihai Putinar, and Igor Mezić. "Data-driven spectral analysis of the Koopman operator". In: *Applied and Computational Harmonic Analysis* 48.2 (2020). Publisher: Elsevier, pp. 599–629.

[215] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. "Neural operator: Learning maps between function spaces". In: *arXiv preprint arXiv:2108.08481* (2021).

[216] Boris P Kovatchev, Marc Breton, Chiara Dalla Man, and Claudio Cobelli. *In silico preclinical trials: a proof of concept in closed-loop control of type 1 diabetes*. 2009.

[217] K. Krischer, R. Rico-Martínez, I. G. Kevrekidis, H. H. Rotermund, G. Ertl, and J. L. Hudson. "Model identification of a spatiotemporally varying catalytic reaction". en. In: *AIChE Journal* 39.1 (Jan. 1993), pp. 89–98. ISSN: 0001-1541, 1547-5905. DOI: 10.1002/aic.690390110. URL: http://doi.wiley.com/10.1002/aic.690390110 (visited on 05/23/2020).

[218] S Krumscheid, M Pradas, GA Pavliotis, and S Kalliadasis. "Data-driven coarse graining in action: Modeling and prediction of complex systems". In: *Physical Review E* 92.4 (2015), p. 042139.

[219] Sebastian Krumscheid, Grigorios A Pavliotis, and Serafim Kalliadasis. "Semiparametric drift and diffusion estimation for multiscale diffusions". In: *Multiscale Modeling & Simulation* 11.2 (2013), pp. 442–473.

[220] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (Mar. 1951). Publisher: Institute of Mathematical Statistics, pp. 79–86. ISSN: 0003-4851, 2168-8990. DOI: 10.1214/aoms/1177729694. URL: https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-22/issue-1/On-Information-and-Sufficiency/10.1214/aoms/1177729694.full (visited on 04/14/2021).

[221] Yury A. Kutoyants. *Statistical Inference for Ergodic Diffusion Processes*. en. Springer Series in Statistics. London: Springer-Verlag, 2004. ISBN: 978-1-85233-759-9. DOI: 10.1007/978-1-4471-3866-2. URL: https://www.springer.com/gp/book/9781852337599 (visited on 06/15/2021).

[222] Steven J Lade. "Finite sampling interval effects in Kramers–Moyal analysis". In: *Physics Letters A* 373.41 (2009), pp. 3705–3709.

[223] I. E. Lagaris, A. Likas, and D. I. Fotiadis. "Artificial Neural Networks for Solving Ordinary and Partial Differential Equations". en. In: *IEEE Transactions on Neural Networks* 9.5 (Sept. 1998). arXiv: physics/9705023, pp. 987–1000. ISSN: 10459227. DOI: 10.1109/72.712178. URL: http://arxiv.org/abs/physics/9705023 (visited on 05/27/2021).

[224] Kody Law, Abhishek Shukla, and Andrew Stuart. "Analysis of the 3DVAR filter for the partially observed Lorenz'63 model". en. In: *Discrete and Continuous Dynamical Systems* 34.3 (Aug. 2013), pp. 1061–1078. ISSN: 1078-0947. DOI: 10.3934/dcds.2014.34.1061. URL: http://www.aimsciences.org/journals/displayArticlesnew.jsp?paperID=8863 (visited on 06/20/2019).

[225] Kody Law, Andrew Stuart, and Kostas Zygalakis. *Data Assimilation*. Springer, 2015.

[226] Kody Law, Andrew Stuart, and Kostas Zygalakis. *Data Assimilation*. Springer, 2015.

[227] Kody Law, Andrew Stuart, and Kostas Zygalakis. "Data assimilation". In: *Cham, Switzerland: Springer* 214 (2015). Publisher: Springer.

[228] Aaron J Le Compte, Dominic S Lee, J Geoffrey Chase, Jessica Lin, Adrienne Lynn, and Geoffrey M Shaw. "Blood glucose prediction using stochastic modeling in neonatal intensive care". In: *IEEE Transactions on Biomedical Engineering* 57.3 (2009), pp. 509–518.

[229] Jonghyeon Lee, Edward De Brouwer, Boumediene Hamzi, and Houman Owhadi. "Learning dynamical systems from data: A simple cross-validation perspective, part iii: Irregularly-sampled time series". In: *Physica D: Nonlinear Phenomena* (2022), p. 133546.

[230] Seunghee Lee, Ahmad S. Khalil, and Wilson W. Wong. "Recent progress of gene circuit designs in immune cell therapies". In: *Cell Systems* 13.11 (2022), pp. 864–873. ISSN: 2405-4712. DOI: https://doi.org/10.1016/j.cels.2022.09.006. URL: https://www.sciencedirect.com/science/article/pii/S240547122200401X.

[231] ED Lehmann and T Deutsch. "A physiological model of glucose-insulin interaction in type 1 diabetes mellitus". In: *Journal of biomedical engineering* 14.3 (1992), pp. 235–242.

[232] Jianjun Lei, Siyuan Liu, and XY Wang. "Dynamic inversion in electrical capacitance tomography using the ensemble Kalman filter". In: *IET Science, Measurement & Technology* 6.2 (2012), pp. 63–77.

[233] Youming Lei, Jian Hu, and Jianpeng Ding. "A hybrid model based on deep LSTM for predicting high-dimensional chaotic systems". en. In: *arXiv:2002.00799 [cs, eess]* (Jan. 2020). arXiv: 2002.00799. URL: http://arxiv.org/abs/2002.00799 (visited on 02/10/2020).

[234] Pablo Lemos, Niall Jeffrey, Miles Cranmer, Shirley Ho, and Peter Battaglia. "Rediscovering orbital mechanics with machine learning". In: (2022).

[235] Matthew Levine, David Albers, and George Hripcsak. "Methodological variations in lagged regression for detecting physiologic drug effects in EHR data". In: *Journal of Biomedical Informatics* 86 (2018), pp. 149–159.

[236] Matthew Levine and Andrew Stuart. "A framework for machine learning of model error in dynamical systems". In: *Communications of the American Mathematical Society* 2.07 (2022), pp. 283–344.

[237] Matthew E. Levine and Andrew M. Stuart. "A Framework for Machine Learning of Model Error in Dynamical Systems". In: *Communications of the American Mathematical Society* 2.07 (2022), pp. 283–344. ISSN: 2692-3688. DOI: 10.1090/cams/10.

[238] ME Levine and AM Stuart. "A Framework for Machine Learning of Model Error in Dynamical Systems". In: *Communications of the American Mathematical Society* 2.07 (2022), pp. 283–344. ISSN: 2692-3688. DOI: 10.1090/cams/10.

[239] Jiaxu Li and Yang Kuang. "Analysis of a model of the glucose-insulin regulatory system with two delays". In: *SIAM Journal on Applied Mathematics* 67.3 (2007), pp. 757–776.

[240] Jiaxu Li, Minghu Wang, Andrea De Gaetano, Pasquale Palumbo, and Simona Panunzi. "The range of time delay and the global stability of the equilibrium for an IVGTT model". In: *Mathematical biosciences* 235.2 (2012), pp. 128–137.

[241] Ruoxia Li, Nabil Magbool Jan, Vinay Prasad, and Biao Huang. "Constrained Extended Kalman Filter based on Kullback-Leibler (KL) Divergence". In: *2018 European Control Conference (ECC)*. IEEE. 2018, pp. 831–836.

[242] Zhen Li, Hee Sun Lee, Eric Darve, and George Em Karniadakis. "Computing the non-Markovian coarse-grained interactions derived from the Mori–Zwanzig formalism in molecular systems: Application to polymer melts". In: *The Journal of Chemical Physics* 146.1 (2017). Publisher: AIP Publishing LLC, p. 014104.

[243] Zhong Li, Jiequn Han, Weinan E, and Qianxiao Li. "On the Curse of Memory in Recurrent Neural Networks: Approximation and Optimization Analysis". In: *arXiv:2009.07799 [cs, math, stat]* (Sept. 2020). arXiv: 2009.07799. URL: http://arxiv.org/abs/2009.07799 (visited on 09/21/2020).

[244] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. "Fourier Neural Operator for Parametric Partial Differential Equations". In: *arXiv:2010.08895 [cs, math]* (Mar. 2021). arXiv: 2010.08895. URL: http://arxiv.org/abs/2010.08895 (visited on 04/22/2021).

[245] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. "Markov Neural Operators for Learning Chaotic Systems". In: *arXiv:2106.06898 [cs, math]* (June 2021). arXiv: 2106.06898. URL: http://arxiv.org/abs/2106.06898 (visited on 08/04/2021).

[246] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, et al. "Fourier Neural Operator for Parametric Partial Differential Equations". In: *International Conference on Learning Representations*. 2020.

[247] Wendell A. Lim. "The emerging era of cell engineering: Harnessing the modularity of cells to program complex biological function". In: *Science* 378.6622 (2022), pp. 848–852. DOI: 10.1126/science.add9665. eprint: https://www.science.org/doi/pdf/10.1126/science.add9665. URL: https://www.science.org/doi/abs/10.1126/science.add9665.

[248] J Lin, JG Chase, GM Shaw, CV Doran, CE Hann, MB Robertson, PM Browne, T Lotz, GC Wake, and B Broughton. "Adaptive bolus-based set-point regulation of hyperglycemia in critical care". In: *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Vol. 2. IEEE. 2004, pp. 3463–3466.

[249] Jessica Lin, Dominic Lee, J Geoffrey Chase, Geoffrey M Shaw, Christopher E Hann, Thomas Lotz, and Jason Wong. "Stochastic modelling of insulin sensitivity variability in critical care". In: *Biomedical Signal Processing and Control* 1.3 (2006), pp. 229–242.

[250] Jessica Lin, Dominic Lee, J Geoffrey Chase, Geoffrey M Shaw, Aaron Le Compte, Thomas Lotz, Jason Wong, Timothy Lonergan, and Christopher E Hann. "Stochastic modelling of insulin sensitivity and adaptive glycemic control for critical care". In: *Computer methods and programs in biomedicine* 89.2 (2008), pp. 141–152.

[251] Jessica Lin, Normy N Razak, Christopher G Pretty, Aaron Le Compte, Paul Docherty, Jacquelyn D Parente, Geoffrey M Shaw, Christopher E Hann, and J Geoffrey Chase. "A physiological Intensive Control Insulin-Nutrition-Glucose (ICING) model validated in critically ill patients". In: *Computer methods and programs in biomedicine* 102.2 (2011), pp. 192–205.

[252] Kevin K Lin and Fei Lu. "Data-driven model reduction, Wiener projections, and the Koopman-Mori-Zwanzig formalism". In: *Journal of Computational Physics* 424 (2021), p. 109864.

[253] Kevin K. Lin and Fei Lu. "Data-driven model reduction, Wiener projections, and the Koopman-Mori-Zwanzig formalism". en. In: *Journal of Computational Physics* 424 (Jan. 2021), p. 109864. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2020.109864. URL: https://www.sciencedirect.com/science/article/pii/S0021999120306380 (visited on 02/09/2021).

[254] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance". In: *Journal of Fluid Mechanics* 807 (2016), pp. 155–166.

[255] Ori Linial, Neta Ravid, Danny Eytan, and Uri Shalit. "Generative ODE modeling with known unknowns". In: *Proceedings of the Conference on Health, Inference, and Learning*. CHIL '21. New York, NY, USA: Association for Computing Machinery, Apr. 2021, pp. 79–94. ISBN: 978-1-4503-8359-2. DOI: 10.1145/3450439.3451866. URL: https://doi.org/10.1145/3450439.3451866 (visited on 04/09/2021).

[256] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. Vol. 793. John Wiley & Sons, 2019.

[257] Chengyuan Liu, Josep Vehı, Nick Oliver, Pantelis Georgiou, and Pau Herrero. "Enhancing Blood Glucose Prediction with Meal Absorption and Physical Exercise Information". In: (), p. 10.

[258] Jun S Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

[259] Weijiu Liu, ChingChun Hsin, and Fusheng Tang. "A molecular mathematical model of glucose mobilization and uptake". In: *Mathematical biosciences* 221.2 (2009), pp. 121–129.

[260] E. Lorenz. "Predictability-A problem partly solved". In: *Proc. Seminar on Predictability, Reading, UK, ECMWF, 1996* (1996). URL: https://ci.nii.ac.jp/naid/10015392260/en/ (visited on 04/09/2021).

[261] Edward N Lorenz. “Predictability: A problem partly solved”. In: *Proc. Seminar on predictability*. Vol. 1. 1. 1996.

[262] Edward N. Lorenz. “Deterministic Nonperiodic Flow”. EN. In: *Journal of the Atmospheric Sciences* 20.2 (Mar. 1963). Publisher: American Meteorological Society Section: Journal of the Atmospheric Sciences, pp. 130–141. ISSN: 0022-4928, 1520-0469. DOI: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2. URL: https://journals.ametsoc.org/view/journals/atsc/20/2/1520-0469_1963_020_0130_dnf_2_0_co_2.xml (visited on 04/09/2021).

[263] Robert J. Lovelett, José L. Avalos, and Ioannis G. Kevrekidis. “Partial Observations and Conservation Laws: Gray-Box Modeling in Biotechnology and Optogenetics”. In: *Industrial & Engineering Chemistry Research* 59.6 (Feb. 2020). Publisher: American Chemical Society, pp. 2611–2620. ISSN: 0888-5885. DOI: 10.1021/acs.iecr.9b04507. URL: https://doi.org/10.1021/acs.iecr.9b04507 (visited on 06/02/2021).

[264] Fei Lu. “Data-Driven Model Reduction for Stochastic Burgers Equations”. en. In: *Entropy* 22.12 (Dec. 2020). Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, p. 1360. DOI: 10.3390/e22121360. URL: https://www.mdpi.com/1099-4300/22/12/1360 (visited on 08/05/2021).

[265] Fei Lu. “Data-driven model reduction for stochastic Burgers equations”. In: *Entropy* 22.12 (2020), p. 1360.

[266] Fei Lu, Kevin Lin, and Alexandre Chorin. “Comparison of continuous and discrete-time data-based modeling for hypoelliptic systems”. In: *Communications in Applied Mathematics and Computational Science* 11.2 (Dec. 2016). Publisher: Mathematical Sciences Publishers, pp. 187–216. ISSN: 2157-5452. DOI: 10.2140/camcos.2016.11.187. URL: https://msp.org/camcos/2016/11-2/p03.xhtml (visited on 08/05/2021).

[267] Fei Lu, Kevin K. Lin, and Alexandre J. Chorin. “Data-based stochastic model reduction for the Kuramoto–Sivashinsky equation”. en. In: *Physica D: Nonlinear Phenomena* 340 (Feb. 2017), pp. 46–57. ISSN: 0167-2789. DOI: 10.1016/j.physd.2016.09.007. URL: https://www.sciencedirect.com/science/article/pii/S0167278915301652 (visited on 08/05/2021).

[268] Lu Lu, Pengzhan Jin, and George Em Karniadakis. “DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators”. In: *arXiv:1910.03193 [cs, stat]* (Apr. 2020). arXiv: 1910.03193. URL: http://arxiv.org/abs/1910.03193 (visited on 10/02/2020).

[269] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. “Learning nonlinear operators via DeepONet based on the universal

approximation theorem of operators". In: *Nature Machine Intelligence* 3.3 (2021), pp. 218–229.

[270] Zhixin Lu, Brian R. Hunt, and Edward Ott. "Attractor reconstruction by machine learning". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.6 (June 2018). Publisher: American Institute of Physics, p. 061104. ISSN: 1054-1500. DOI: 10.1063/1.5039508. URL: https://aip.scitation.org/doi/10.1063/1.5039508 (visited on 06/02/2021).

[271] Valerio Lucarini, Richard Blender, Corentin Herbert, Francesco Ragone, Salvatore Pascale, and Jeroen Wouters. "Mathematical and physical ideas for climate science". In: *Rev. Geophys.* 52 (2014), pp. 809–859.

[272] Mantas Lukoševičius and Herbert Jaeger. "Reservoir computing approaches to recurrent neural network training". en. In: *Computer Science Review* 3.3 (Aug. 2009), pp. 127–149. ISSN: 1574-0137. DOI: 10.1016/j.cosrev.2009.03.005. URL: https://www.sciencedirect.com/science/article/pii/S1574013709000173 (visited on 06/21/2021).

[273] Chao Ma, Jianchun Wang, et al. "Model reduction with memory and the machine learning of dynamical systems". In: *arXiv preprint arXiv:1808.04258* (2018).

[274] Chao Ma, Jianchun Wang, and Weinan E. "Model Reduction with Memory and the Machine Learning of Dynamical Systems". In: *Communications in Computational Physics* 25.4 (2019). ISSN: 18152406. DOI: 10.4208/cicp.OA-2018-0269. URL: http://www.global-sci.com/intro/article_detail/cicp/12885.html (visited on 05/07/2021).

[275] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[276] Lena Mamykina, Matthew E Levine, Patricia G Davidson, Arlene M Smaldone, Noemie Elhadad, and David J Albers. "Data-driven health management: reasoning about personally generated data in diabetes with information technologies". In: *Journal of the American Medical Informatics Association* 23.3 (2016), pp. 526–531.

[277] Lena Mamykina, Matthew E Levine, Patricia G Davidson, Arlene M Smaldone, Noemie Elhadad, and David J Albers. "From personal informatics to personal analytics: Investigating how clinicians and patients reason about personal data generated with self-monitoring in diabetes". In: *Cognitive Informatics in Health and Biomedicine*. Springer, Cham, 2017, pp. 301–313.

[278] RK Mandela, V Kuppuraj, R Rengaswamy, and S Narasimhan. "Constrained unscented recursive estimator for nonlinear dynamic systems". In: *Journal of Process Control* 22.4 (2012), pp. 718–728.

[279] Andrea Mari, Andrea Tura, Eleonora Grespan, and Roberto Bizzotto. “Mathematical Modeling for the Physiological and Clinical Investigation of Glucose Homeostasis and Diabetes”. In: *Frontiers in Physiology* 11 (Nov. 2020), p. 575789. ISSN: 1664-042X. DOI: 10.3389/fphys.2020.575789.

[280] Andrea Mari, Andrea Tura, Eleonora Grespan, and Roberto Bizzotto. “Mathematical modeling for the physiological and clinical investigation of glucose homeostasis and diabetes”. In: *Frontiers in Physiology* 11 (2020), p. 1548.

[281] Barbara Martinovic, John Leth, Torben Knudsen, Tinna Björk Aradóttir, and Henrik Bengtsson. “Modelling the glucose-insulin system of type 2 diabetes patients using ARMAX models”. In: *2019 Australian & New Zealand Control Conference (ANZCC)*. IEEE. 2019, pp. 88–93.

[282] Romit Maulik, Bethany Lusch, and Prasanna Balaprakash. “Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders”. In: *Physics of Fluids* 33.3 (Mar. 2021). Publisher: American Institute of Physics, p. 037106. ISSN: 1070-6631. DOI: 10.1063/5.0039986. URL: https://aip.scitation.org/doi/abs/10.1063/5.0039986 (visited on 04/09/2021).

[283] Romit Maulik, Omer San, Adil Rasheed, and Prakash Vedula. “Subgrid modelling for two-dimensional turbulence using neural networks”. In: *Journal of Fluid Mechanics* 858 (2019), pp. 122–144.

[284] Kevin McGoff, Sayan Mukherjee, Andrew Nobel, and Natesh Pillai. “Consistency of maximum likelihood estimation for some dynamical systems”. In: *The Annals of Statistics* 43.1 (Feb. 2015). Publisher: Institute of Mathematical Statistics, pp. 1–29. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/14-AOS1259. URL: https://projecteuclid.org/journals/annals-of-statistics/volume-43/issue-1/Consistency-of-maximum-likelihood-estimation-for-some-dynamical-systems/10.1214/14-AOS1259.full (visited on 06/15/2021).

[285] Xiao-Li Meng and David Van Dyk. “The EM algorithm—an old folk-song sung to a fast new tune”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 59.3 (1997), pp. 511–567.

[286] Andrew C Miller, Nicholas J Foti, and Emily Fox. “Learning Insulin-Glucose Dynamics in the Wild”. In: *Machine Learning for Healthcare Conference*. PMLR. 2020, pp. 172–197.

[287] Andrew C. Miller, Nicholas J. Foti, and Emily Fox. “Learning Insulin-Glucose Dynamics in the Wild”. In: *arXiv:2008.02852 [cs, stat]* (Aug. 2020). arXiv: 2008.02852. URL: http://arxiv.org/abs/2008.02852 (visited on 09/15/2020).

[288] Andrew C. Miller, Nicholas J. Foti, and Emily B. Fox. “Breiman’s Two Cultures: You Don’t Have to Choose Sides”. In: *Observational Studies* 7.1 (2021), pp. 161–169. ISSN: 2767-3324. DOI: 10.1353/obs.2021.0003.

[289] Andrew C. Miller, Nicholas J. Foti, and Emily B. Fox. “Breiman’s two cultures: You don’t have to choose sides”. In: *arXiv:2104.12219 [cs, stat]* (Apr. 2021). arXiv: 2104.12219. URL: http://arxiv.org/abs/2104.12219 (visited on 07/09/2021).

[290] Elliot G Mitchell, Esteban G Tabak, Matthew E Levine, Lena Mamykina, and David J Albers. “Enabling personalized decision support with patient-generated data and attributable components”. In: *Journal of Biomedical Informatics* 113 (2021), p. 103639.

[291] Jonas Mockus. *Bayesian Approach to Global Optimization: Theory and Applications*. en. Mathematics and its Applications. Springer Netherlands, 1989. ISBN: 978-94-010-6898-7. DOI: 10.1007/978-94-009-0909-0. URL: https://www.springer.com/gp/book/9789401068987 (visited on 06/16/2021).

[292] Eslam Montaser, José-Luis Díez, and Jorge Bondia. “Stochastic seasonal models for glucose prediction in the artificial pancreas”. In: *Journal of diabetes science and technology* 11.6 (2017), pp. 1124–1131.

[293] Glen H Murata, Richard M Hoffman, Jayendra H Shah, Christopher S Wendel, and William C Duckworth. “A probabilistic model for predicting hypoglycemia in type 2 diabetes mellitus: The Diabetes Outcomes in Veterans Study (DOVES)”. In: *Archives of internal medicine* 164.13 (2004), pp. 1445–1450.

[294] Kumpati S. Narendra and Kannan Parthasarathy. “Neural networks and dynamical systems”. en. In: *International Journal of Approximate Reasoning* 6.2 (Feb. 1992), pp. 109–131. ISSN: 0888-613X. DOI: 10.1016/0888-613X(92)90014-Q. URL: https://www.sciencedirect.com/science/article/pii/0888613X9290014Q (visited on 04/22/2021).

[295] Nicholas H. Nelsen and Andrew M. Stuart. “The Random Feature Model for Input-Output Maps between Banach Spaces”. In: *arXiv:2005.10224 [physics, stat]* (May 2020). arXiv: 2005.10224. URL: http://arxiv.org/abs/2005.10224 (visited on 04/08/2021).

[296] Nicholas H. Nelsen and Andrew M. Stuart. “The Random Feature Model for Input-Output Maps between Banach Spaces”. In: *SIAM Journal on Scientific Computing* 43.5 (2021), A3212–A3243. eprint: https://doi.org/10.1137/20M133957X.

[297] Arnold Neumaier and Tapio Schneider. “Estimation of parameters and eigenmodes of multivariate autoregressive models”. In: *ACM Transactions on Mathematical Software (TOMS)* 27.1 (2001), pp. 27–57.

[298] Duong Nguyen, Said Ouala, Lucas Drumetz, and Ronan Fablet. “EM-like Learning Chaotic Dynamics from Noisy and Partial Observations”. In: *arXiv:1903.10335 [cs, stat]* (Mar. 2019). arXiv: 1903.10335. URL: http://arxiv.org/abs/1903.10335 (visited on 12/05/2020).

[299] Murphy Yuezhen Niu, Lior Horesh, and Isaac Chuang. “Recurrent Neural Networks in the Eye of Differential Equations”. In: *arXiv:1904.12933 [quant-ph, stat]* (Apr. 2019). arXiv: 1904.12933. URL: http://arxiv.org/abs/1904.12933 (visited on 09/10/2020).

[300] Fernando Nogueira. *Bayesian Optimization: Open source constrained global optimization tool for Python*. 2014. URL: https://github.com/fmfn/BayesianOptimization.

[301] Paul A. O’Gorman and John G. Dwyer. “Using Machine Learning to Parameterize Moist Convection: Potential for Modeling of Climate, Climate Change, and Extreme Events”. en. In: *Journal of Advances in Modeling Earth Systems* 10.10 (2018). _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2018MS001351, pp. 2548–2563. ISSN: 1942-2466. DOI: https://doi.org/10.1029/2018MS001351. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018MS001351 (visited on 04/09/2021).

[302] Dean S Oliver, Albert C Reynolds, and Ning Liu. *Inverse theory for petroleum reservoir characterization and history matching*. Cambridge University Press, 2008.

[303] Todd A Oliver and Robert D Moser. “Bayesian uncertainty quantification applied to RANS turbulence models”. In: *Journal of Physics: Conference Series*. Vol. 318. IOP Publishing. 2011, p. 042032.

[304] S. Ouala, D. Nguyen, L. Drumetz, B. Chapron, A. Pascual, F. Collard, L. Gaultier, and R. Fablet. “Learning latent dynamics for partially observed chaotic systems”. en. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.10 (Oct. 2020), p. 103121. ISSN: 1054-1500, 1089-7682. DOI: 10.1063/5.0019309. URL: http://aip.scitation.org/doi/10.1063/5.0019309 (visited on 08/02/2021).

[305] Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[306] TN Palmer. “Stochastic weather and climate models”. In: *Nature Reviews Physics* 1.7 (2019), pp. 463–471.

[307] Guofei Pang, Marta D’Elia, Michael Parks, and George E Karniadakis. “nPINNs: nonlocal Physics-Informed Neural Networks for a parametrized nonlocal universal Laplacian operator. Algorithms and Applications”. In: *Journal of Computational Physics* 422 (2020), p. 109760.

[308] Omiros Papaspiliopoulos, Yvo Pokern, Gareth O Roberts, and Andrew M Stuart. “Nonparametric estimation of diffusions: a differential equations approach”. In: *Biometrika* 99.3 (2012), pp. 511–531.

[309] Omiros Papaspiliopoulos, Gareth O Roberts, and Osnat Stramer. “Data augmentation for diffusions”. In: *Journal of Computational and Graphical Statistics* 22.3 (2013), pp. 665–688.

[310] A Papavasiliou, GA Pavliotis, and AM Stuart. “Maximum likelihood drift estimation for multiscale diffusions”. In: *Stochastic Processes and their Applications* 119.10 (2009), pp. 3173–3210.

[311] Eric J Parish and Karthik Duraisamy. “A dynamic subgrid scale model for large eddy simulations based on the Mori–Zwanzig formalism”. In: *Journal of Computational Physics* 349 (2017). Publisher: Elsevier, pp. 154–175.

[312] Robert S Parker, Francis J Doyle, and Nicholas A Peppas. “The intravenous route to blood glucose control”. In: *IEEE Engineering in Medicine and Biology Magazine* 20.1 (2001), pp. 65–73.

[313] Robert S Parker, Francis J Doyle, Jennifer H Ward, and Nicholas A Peppas. “Robust H glucose control in diabetes using a physiological model”. In: *AIChE Journal* 46.12 (2000), pp. 2537–2549.

[314] Robert S Parker, Timothy D Knab, and Gilles Clermont. “The Impact of a Responsive Endogenous Pancreas in Critical Care Glucose Control”. In: *2018 Annual American Control Conference (ACC)*. IEEE. 2018, pp. 3595–3601.

[315] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. “Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach”. en. In: *Physical Review Letters* 120.2 (Jan. 2018), p. 024102. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.120.024102. URL: https://link.aps.org/doi/10.1103/PhysRevLett.120.024102 (visited on 06/23/2021).

[316] Jaideep Pathak, Alexander Wikner, Rebeckah Fussell, Sarthak Chandra, Brian R. Hunt, Michelle Girvan, and Edward Ott. “Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 28.4 (Apr. 2018), p. 041101. ISSN: 1054-1500. DOI: 10.1063/1.5028373. URL: https://aip.scitation.org/doi/10.1063/1.5028373 (visited on 02/04/2019).

[317] Grigorios A Pavliotis, Yvo Pokern, and Andrew M Stuart. “Parameter estimation for multiscale diffusions: An overview”. In: *Statistical Methods for Stochastic Differential Equations* (2012), p. 429.

[318] Grigorios A Pavliotis and AM Stuart. “Parameter estimation for multiscale diffusions”. In: *Journal of Statistical Physics* 127.4 (2007), pp. 741–781.

[319] Grigoris Pavliotis and Andrew Stuart. *Multiscale methods: averaging and homogenization*. Springer Science & Business Media, 2008.

[320] Cecile Penland and Theresa Magorian. “Prediction of Nino 3 sea surface temperatures using linear inverse modeling”. In: *Journal of Climate* 6.6 (1993), pp. 1067–1076.

[321] Pascal Pernot and Fabien Cailliez. “A critical review of statistical calibration/prediction models handling data inconsistency and model inadequacy”. In: *AIChE Journal* 63.10 (2017), pp. 4642–4665.

[322] Rimma Pivovarov, David J Albers, Jorge L Sepulveda, and Noémie Elhadad. “Identifying and mitigating biases in EHR laboratory tests”. In: *Journal of biomedical informatics* 51 (2014), pp. 24–34.

[323] Matthew Plumlee. “Bayesian Calibration of Inexact Computer Models”. en. In: *Journal of the American Statistical Association* 112.519 (July 2017), pp. 1274–1285. ISSN: 0162-1459, 1537-274X. DOI: 10.1080/01621459.2016.1211016. URL: https://www.tandfonline.com/doi/full/10.1080/01621459.2016.1211016 (visited on 05/04/2021).

[324] Matthew Plumlee and V. Roshan Joseph. “Orthogonal Gaussian process models”. en. In: *Statistica Sinica* (2017). ISSN: 10170405. DOI: 10.5705/ss.202015.0404. URL: http://www3.stat.sinica.edu.tw/statistica/J28N2/J28N23/J28N23.html (visited on 05/03/2021).

[325] Yvo Pokern, Andrew M Stuart, and Petter Wiberg. “Parameter estimation for partially observed hypoelliptic diffusions”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 71.1 (2009), pp. 49–73.

[326] J Prakash, Sachin C Patwardhan, and Sirish L Shah. “Constrained nonlinear state estimation using ensemble Kalman filters”. In: *Industrial & Engineering Chemistry Research* 49.5 (2010), pp. 2242–2253.

[327] Jagdeesan Prakash, S. C. Patwardhan, and S. L. Shah. “Constrained state estimation using the ensemble Kalman filter”. In: *2008 American Control Conference* (2008), pp. 3542–3547.

[328] Ari Pritchard-Bell, Gilles Clermont, Timothy D Knab, John Maalouf, Michael Vilkhovoy, and Robert S Parker. “Modeling glucose and subcutaneous insulin dynamics in critical care”. In: *Control Engineering Practice* 58 (2017), pp. 268–275.

[329] Manuel Pulido, Pierre Tandeo, Marc Bocquet, Alberto Carrassi, and Magdalena Lucini. “Stochastic parameterization identification using ensemble Kalman filtering combined with maximum likelihood methods”. In: *Tellus A: Dynamic Meteorology and Oceanography* 70.1 (2018). Publisher: Taylor & Francis, pp. 1–17.

[330] Ryan Pyle, Nikola Jovanovic, Devika Subramanian, Krishna V. Palem, and Ankit B. Patel. “Domain-driven models yield better predictions at lower cost than reservoir computers in Lorenz systems”. en. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379.2194 (Apr. 2021), p. 20200246. ISSN: 1364-503X, 1471-2962. DOI: 10.1098/rsta.2020.0246. URL: https://royalsocietypublishing.org/doi/10.1098/rsta.2020.0246 (visited on 08/02/2021).

[331] Zhaozhi Qian, William R. Zame, Lucas M. Fleuren, Paul Elbers, and Mihaela van der Schaar. "Integrating Expert ODEs into Neural ODEs: Pharmacology and Disease Progression". In: *arXiv:2106.02875 [cs, stat]* (June 2021). arXiv: 2106.02875. URL: http://arxiv.org/abs/2106.02875 (visited on 06/16/2021).

[332] Alejandro F. Queiruga, N. Benjamin Erichson, Dane Taylor, and Michael W. Mahoney. "Continuous-in-Depth Neural Networks". In: *arXiv:2008.02389 [cs, math, stat]* (Aug. 2020). arXiv: 2008.02389. URL: http://arxiv.org/abs/2008.02389 (visited on 09/21/2020).

[333] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. "Universal Differential Equations for Scientific Machine Learning". In: *arXiv:2001.04385 [cs, math, q-bio, stat]* (Aug. 2020). arXiv: 2001.04385. URL: http://arxiv.org/abs/2001.04385 (visited on 06/16/2021).

[334] Christopher Rackauckas, Roshan Sharma, and Bernt Lie. "Hybrid Mechanistic + Neural Model of Laboratory Helicopter". en. In: Mar. 2021, pp. 264–271. DOI: 10.3384/ecp20176264. URL: https://ep.liu.se/en/conference-article.aspx?series=ecp&issue=176&Article_No=37 (visited on 06/16/2021).

[335] Ali Rahimi and Benjamin Recht. "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems*. Ed. by J. Platt, D. Koller, Y. Singer, and S. Roweis. Vol. 20. Curran Associates, Inc., 2008. URL: https://proceedings.neurips.cc/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

[336] Ali Rahimi and Benjamin Recht. "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems*. Ed. by J. Platt, D. Koller, Y. Singer, and S. Roweis. Vol. 20. Curran Associates, Inc., 2008.

[337] Ali Rahimi and Benjamin Recht. "Uniform approximation of functions with random bases". In: *2008 46th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2008, pp. 555–561.

[338] Ali Rahimi and Benjamin Recht. "Weighted sums of random kitchen sinks: replacing minimization with randomization in learning." In: *Nips*. Citeseer, 2008, pp. 1313–1320.

[339] M. Raissi, P. Perdikaris, and G. E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". en. In: *Journal of Computational Physics* 378 (Feb. 2019), pp. 686–707. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2018.10.045. URL: https://www.sciencedirect.com/science/article/pii/S0021999118307125 (visited on 04/06/2021).

[340] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. “Multistep Neural Networks for Data-driven Discovery of Nonlinear Dynamical Systems”. en. In: *arXiv:1801.01236 [nlin, physics:physics, stat]* (Jan. 2018). arXiv: 1801.01236. URL: http://arxiv.org/abs/1801.01236 (visited on 11/27/2019).

[341] Christopher V Rao, James B Rawlings, and David Q Mayne. “Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations”. In: *IEEE transactions on automatic control* 48.2 (2003), pp. 246–258.

[342] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. en. Adaptive computation and machine learning. OCLC: ocm61285753. Cambridge, Mass: MIT Press, 2006. ISBN: 978-0-262-18253-9.

[343] Stephan Rasp, Michael S. Pritchard, and Pierre Gentine. “Deep learning to represent subgrid processes in climate models”. In: *Proc. Natl. Acad. Sci.* (2018).

[344] Stephan Rasp, Michael S. Pritchard, and Pierre Gentine. “Deep learning to represent subgrid processes in climate models”. en. In: *Proceedings of the National Academy of Sciences* 115.39 (Sept. 2018). ISBN: 9781810286112 Publisher: National Academy of Sciences Section: Physical Sciences, pp. 9684–9689. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1810286115. URL: https://www.pnas.org/content/115/39/9684 (visited on 04/09/2021).

[345] Sebastian Reich and Colin Cotter. *Probabilistic forecasting and Bayesian data assimilation*. Cambridge University Press, 2015.

[346] Sebastian Reich and Colin Cotter. *Probabilistic forecasting and Bayesian data assimilation*. Cambridge University Press, 2015.

[347] Jaques Reifman, Srinivasan Rajaraman, Andrei Gribok, and W Kenneth Ward. “Predictive monitoring for improved management of glucose levels”. In: *Journal of diabetes science and technology* 1.4 (2007), pp. 478–486.

[348] R. Rico-Martines, I. G. Kevrekidis, M. C. Kube, and J. L. Hudson. “Discrete-vs. Continuous-Time Nonlinear Signal Processing: Attractors, Transitions and Parallel Implementation Issues”. en. In: *1993 American Control Conference*. San Francisco, CA, USA: IEEE, June 1993, pp. 1475–1479. ISBN: 978-0-7803-0860-2. DOI: 10.23919/ACC.1993.4793116. URL: https://ieeexplore.ieee.org/document/4793116/ (visited on 05/23/2020).

[349] R. Rico-Martinez, J.S. Anderson, and I.G. 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*. Ermioni, Greece: IEEE, 1994, pp. 596–605. ISBN: 978-0-7803-2026-0. DOI: 10.1109/NNSP.1994.366006.

[350] R. Rico-Martinez, J.S. Anderson, and I.G. Kevrekidis. “Continuous-time nonlinear signal processing: a neural network based approach for gray box identification”. en. In: *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*. Ermioni, Greece: IEEE, 1994, pp. 596–605. ISBN: 978-0-7803-2026-0. DOI: 10.1109/NNSP.1994.366006. URL: http://ieeexplore.ieee.org/document/366006/ (visited on 05/23/2020).

[351] R. Rico-Martínez, K. Krischer, I.G. Kevrekidis, M.C. Kube, and J.L. Hudson. “DISCRETE- vs. CONTINUOUS-TIME NONLINEAR SIGNAL PROCESSING OF Cu ELECTRODISSOLUTION DATA”. en. In: *Chemical Engineering Communications* 118.1 (Nov. 1992), pp. 25–48. ISSN: 0098-6445, 1563-5201. DOI: 10.1080/00986449208936084. URL: https://www.tandfonline.com/doi/full/10.1080/00986449208936084 (visited on 05/23/2020).

[352] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[353] Gareth O Roberts and Osnat Stramer. “Langevin diffusions and Metropolis-Hastings algorithms”. In: *Methodology and computing in applied probability* 4.4 (2002), pp. 337–357.

[354] Gareth O Roberts and Osnat Stramer. “On inference for partially observed nonlinear diffusion models using the Metropolis–Hastings algorithm”. In: *Biometrika* 88.3 (2001), pp. 603–621.

[355] Douglas G Robertson, Jay H Lee, and James B Rawlings. “A moving horizon-based approach for least-squares estimation”. In: *AIChE Journal* 42.8 (1996), pp. 2209–2224.

[356] Derrick K Rollins, Nidhi Bhandari, Jim Kleinedler, Kaylee Kotz, Amber Strohbehn, Lindsay Boland, Megan Murphy, Dave Andre, Nisarg Vyas, Greg Welk, et al. “Free-living inferential modeling of blood glucose level using only noninvasive inputs”. In: *Journal of process control* 20.1 (2010), pp. 95–107.

[357] Anirban Roy and Robert S Parker. “A phenomenological model of plasma FFA, glucose, and insulin concentrations during rest and exercise”. In: *Proceedings of the 2010 American Control Conference*. IEEE. 2010, pp. 5161–5166.

[358] Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. “Latent ODEs for Irregularly-Sampled Time Series”. In: *arXiv:1907.03907 [cs, stat]* (July 2019). arXiv: 1907.03907. URL: http://arxiv.org/abs/1907.03907 (visited on 10/12/2020).

[359] Yulia Rubanova, Ricky T. Q. Chen, and David K Duvenaud. “Latent Ordinary Differential Equations for Irregularly-Sampled Time Series”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[360] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. "Data-driven discovery of partial differential equations". In: *Science Advances* 3.4 (2017), e1602614.

[361] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986). Publisher: Nature Publishing Group, pp. 533–536.

[362] Khachik Sargsyan, Xun Huan, and Habib N Najm. "Embedded model error representation for Bayesian model calibration". In: *International Journal for Uncertainty Quantification* 9.4 (2019).

[363] Tim Sauer, James A Yorke, and Martin Casdagli. "Embedology". In: *Journal of statistical Physics* 65.3 (1991), pp. 579–616.

[364] 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)*. ISSN: 2153-0866. Sept. 2017, pp. 4709–4715. DOI: 10.1109/IROS.2017.8206343.

[365] Hayden Schaeffer. "Learning partial differential equations via data discovery and sparse optimization". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473.2197 (2017), p. 20160446.

[366] Hayden Schaeffer, Russel Caflisch, Cory D Hauck, and Stanley Osher. "Sparse dynamics for partial differential equations". In: *Proceedings of the National Academy of Sciences* 110.17 (2013), pp. 6634–6639.

[367] Hayden Schaeffer, Giang Tran, and Rachel Ward. "Extracting sparse high-dimensional dynamics from limited data". In: *SIAM Journal on Applied Mathematics* 78.6 (2018). Publisher: SIAM, pp. 3279–3295.

[368] Hayden Schaeffer, Giang Tran, and Rachel Ward. "Learning dynamical systems and bifurcation via group sparsity". In: *arXiv preprint arXiv:1709.01558* (2017).

[369] Hayden Schaeffer, Giang Tran, Rachel Ward, and Linan Zhang. "Extracting structured dynamical systems using sparse optimization with very few samples". In: *Multiscale Modeling & Simulation* 18.4 (2020). Publisher: SIAM, pp. 1435–1461.

[370] Anton Maximilian Schäfer and Hans-Georg Zimmermann. "Recurrent Neural Networks are universal approximators". eng. In: *International Journal of Neural Systems* 17.4 (Aug. 2007), pp. 253–263. ISSN: 0129-0657. DOI: 10.1142/S0129065707001111.

[371] Robert E. Schapire. "The Strength of Weak Learnability". In: *Machine Learning* 5.2 (June 1990), pp. 197–227. ISSN: 1573-0565. DOI: 10.1007/BF00116037.

[372] Martin Schmelzer, Richard P Dwight, and Paola Cinnella. "Discovery of algebraic Reynolds-stress models using sparse symbolic regression". In: *Flow, Turbulence and Combustion* 104.2 (2020), pp. 579–603.

[373] Tapio Schneider, Shiwei Lan, Andrew Stuart, and Joao Teixeira. "Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations". In: *Geophysical Research Letters* 44.24 (2017), pp. 12–396.

[374] Tapio Schneider, Andrew M Stuart, and Jin-Long Wu. "Ensemble Kalman inversion for sparse learning of dynamical systems from time-averaged data". In: *arXiv preprint arXiv:2007.06175* (2020).

[375] Tapio Schneider, Andrew M Stuart, and Jin-Long Wu. "Learning stochastic closures using ensemble Kalman inversion". In: *Transactions of Mathematics and Its Applications* 5.1 (2021), tnab003.

[376] Tapio Schneider, Andrew M Stuart, and Jin-Long Wu. "Learning stochastic closures using ensemble Kalman inversion". In: *Transactions of Mathematics and Its Applications* 5.1 (2021), tnab003.

[377] Tapio Schneider, Joao Teixeira, Christopher S. Bretherton, Florent Brient, Kyle G. Pressel, Christoph Schar, and A. Pier Siebesma. "Climate goals and computing the future of clouds". In: *Nature Climate Change* 7 (2017), pp. 3–5.

[378] Skipper Seabold and Josef Perktold. "statsmodels: Econometric and statistical modeling with python". In: *9th Python in Science Conference*. 2010.

[379] Madineh Sedigh-Sarvestani, David J Albers, and Bruce J Gluckman. "Data assimilation of glucose dynamics for use in the intensive care unit". In: *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2012, pp. 5437–5440.

[380] Rachel E Sherman, Steven A Anderson, Gerald J Dal Pan, Gerry W Gray, Thomas Gross, Nina L Hunter, Lisa LaVange, Danica Marinac-Dabic, Peter W Marks, Melissa A Robb, et al. "Real-world evidence—what is it and what can it tell us". In: *N Engl J Med* 375.23 (2016), pp. 2293–2297.

[381] Alex Sherstinsky. "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network". en. In: *Physica D: Nonlinear Phenomena* 404 (Mar. 2020), p. 132306. ISSN: 0167-2789. DOI: 10.1016/j.physd.2019.132306. URL: http://www.sciencedirect.com/science/article/pii/S0167278919305974 (visited on 09/01/2020).

[382] 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: *arXiv:1811.08027 [cs]* (Nov. 2018). arXiv: 1811.08027. URL: http://arxiv.org/abs/1811.08027 (visited on 06/07/2019).

[383] 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)*. IEEE. 2019, pp. 9784–9790.

[384] Xiangyun Shi, Qi Zheng, Jiaoyan Yao, Jiaxu Li, and Xueyong Zhou. "Analysis of a stochastic IVGTT glucose-insulin model with time delay". In: *Mathematical Biosciences and Engineering* 17.3 (2020), pp. 2310–2329.

[385] B. M. de Silva, D. M. Higdon, S. L. Brunton, and J. N. Kutz. "Discovery of Physics From Data: Universal Laws and Discrepancies". In: *Front. Artif. Intell.* 3 (2020), p. 25.

[386] Dan Simon. "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms". In: *IET Control Theory & Applications* 4.8 (2010), pp. 1303–1318.

[387] Dan Simon and Donald L Simon. "Constrained Kalman filtering via density function truncation for turbofan engine health estimation". In: *International Journal of Systems Science* 41.2 (2010), pp. 159–171.

[388] Dan Simon and Donald L Simon. "Kalman filtering with inequality constraints for turbofan engine health estimation". In: *IEE Proceedings-Control Theory and Applications* 153.3 (2006), pp. 371–378.

[389] M. Sirlanci, M. E. Levine, C. C. Low Wang, D. J. Albers, and A. M. Stuart. "A Simple Modeling Framework For Prediction In The Human Glucose-Insulin System". In: *Chaos (To appear)* (2023). arXiv: 1910.14193 [q-bio.QM].

[390] J. Smagorinsky. "General Circulation Experiments with the Primitive Equations. I. The Basic Experiment". In: *Mon. Wea. Rev.* 91 (1963), pp. 99–164.

[391] Jonathan D. Smith, Kamyar Azizzadenesheli, and Zachary E. Ross. "EikoNet: Solving the Eikonal Equation With Deep Neural Networks". In: *IEEE Transactions on Geoscience and Remote Sensing* (2020). Conference Name: IEEE Transactions on Geoscience and Remote Sensing, pp. 1–12. ISSN: 1558-0644. DOI: 10.1109/TGRS.2020.3039165.

[392] Peter D. Sottile et al. "Real-Time Electronic Health Record Mortality Prediction During the COVID-19 Pandemic: A Prospective Cohort Study". en. In: *medRxiv* (Jan. 2021). Publisher: Cold Spring Harbor Laboratory Press, p. 2021.01.14.21249793. DOI: 10.1101/2021.01.14.21249793. URL: https://www.medrxiv.org/content/10.1101/2021.01.14.21249793v1 (visited on 04/06/2021).

[393] Giovanni Sparacino, Francesca Zanderigo, Stefano Corazza, Alberto Maran, Andrea Facchinetti, and Claudio Cobelli. "Glucose concentration can be predicted ahead in time from continuous glucose monitoring sensor time-

series". In: *IEEE Transactions on biomedical engineering* 54.5 (2007), pp. 931–937.

[394] Mark Strong and Jeremy E Oakley. "When is a model good enough? Deriving the expected value of model improvement via specifying internal model discrepancies". In: *SIAM/ASA Journal on Uncertainty Quantification* 2.1 (2014), pp. 106–125.

[395] Walter W Stroup. *Generalized linear mixed models: modern concepts, methods and applications*. CRC press, 2012.

[396] Andrew M Stuart. "Inverse problems: a Bayesian perspective". In: *Acta Numerica* 19 (2010), pp. 451–559.

[397] Jeppe Sturis, Kenneth S Polonsky, Erik Mosekilde, and Eve Van Cauter. "Computer Model for Mechanisms Underlying Ultradian Oscillations of Insulin and Glucose". In: *American Journal of Physiology-Endocrinology And Metabolism* 260.5 (1991), E801–E809.

[398] Jeppe Sturis, Kenneth S Polonsky, Erik Mosekilde, and Eve Van Cauter. "Computer model for mechanisms underlying ultradian oscillations of insulin and glucose". In: *American Journal of Physiology-Endocrinology And Metabolism* 260.5 (1991). Publisher: American Physiological Society Bethesda, MD, E801–E809.

[399] Jeppe Sturis, Kenneth S Polonsky, Erik Mosekilde, and Eve Van Cauter. "Computer model for mechanisms underlying ultradian oscillations of insulin and glucose". In: *American Journal of Physiology-Endocrinology And Metabolism* 260.5 (1991), E801–E809.

[400] Langxuan Su and Sayan Mukherjee. "A Large Deviation Approach to Posterior Consistency in Dynamical Systems". In: *arXiv:2106.06894 [math, stat]* (June 2021). arXiv: 2106.06894. URL: http://arxiv.org/abs/2106.06894 (visited on 06/16/2021).

[401] Bharath Sudharsan, Malinda Peeples, and Mansur Shomali. "Hypoglycemia prediction using machine learning models for patients with type 2 diabetes". In: *Journal of diabetes science and technology* 9.1 (2014), pp. 86–90.

[402] Floris Takens. "Detecting strange attractors in turbulence". en. In: *Dynamical Systems and Turbulence, Warwick 1980*. Ed. by David Rand and Lai-Sang Young. Lecture Notes in Mathematics. Berlin, Heidelberg: Springer, 1981, pp. 366–381. ISBN: 978-3-540-38945-3. DOI: 10.1007/BFb0091924.

[403] Zhihong Tan, Colleen M Kaul, Kyle G Pressel, Yair Cohen, Tapio Schneider, and João Teixeira. "An extended eddy-diffusivity mass-flux scheme for unified representation of subgrid-scale turbulence and convection". In: *Journal of Advances in Modeling Earth Systems* 10.3 (2018). Publisher: Wiley Online Library, pp. 770–800.

[404] Martin Tauschmann, Hood Thabit, Lia Bally, Janet M Allen, Sara Hartnell, Malgorzata E Wilinska, Yue Ruan, Judy Sibayan, Craig Kollman, Peiyao Cheng, et al. “Closed-loop insulin delivery in suboptimally controlled type 1 diabetes: a multicentre, 12-week randomised trial”. In: *The Lancet* 392.10155 (2018), pp. 1321–1329.

[405] Bruno OS Teixeira, Leonardo AB Tôrres, Luis A Aguirre, and Dennis S Bernstein. “On unscented Kalman filtering with state interval constraints”. In: *Journal of Process Control* 20.1 (2010), pp. 45–57.

[406] Hood Thabit, Martin Tauschmann, Janet M Allen, Lalantha Leelarathna, Sara Hartnell, Malgorzata E Wilinska, Carlo L Acerini, Sibylle Dellweg, Carsten Benesch, Lutz Heinemann, et al. “Home use of an artificial beta cell in type 1 diabetes”. In: *New England Journal of Medicine* 373.22 (2015), pp. 2129–2140.

[407] Brian Topp, Keith Promislow, Gerda Devries, Robert M Miura, and DIANE T FINEGOOD. “A model of β-cell mass, insulin, and glucose kinetics: pathways to diabetes”. In: *Journal of theoretical biology* 206.4 (2000), pp. 605–619.

[408] Giang Tran and Rachel Ward. “Exact recovery of chaotic systems from highly corrupted data”. In: *Multiscale Modeling & Simulation* 15.3 (2017). Publisher: SIAM, pp. 1108–1129.

[409] Sumeet Trehan, Kevin T Carlberg, and Louis J Durlofsky. “Error modeling for surrogates of dynamical systems using machine learning”. In: *International Journal for Numerical Methods in Engineering* 112.12 (2017), pp. 1801–1827.

[410] Jonathan H. Tu, Clarence W. Rowley, Dirk M. Luchtenburg, Steven L. Brunton, and J. Nathan Kutz. “On dynamic mode decomposition: Theory and applications”. en. In: *Journal of Computational Dynamics* 1.2 (2014). Company: Journal of Computational Dynamics Distributor: Journal of Computational Dynamics Institution: Journal of Computational Dynamics Label: Journal of Computational Dynamics Publisher: American Institute of Mathematical Sciences, p. 391. DOI: 10.3934/jcd.2014.1.391. URL: https://www.aimsciences.org/article/doi/10.3934/jcd.2014.1.391 (visited on 04/22/2021).

[411] Pramod Vachhani, Shankar Narasimhan, and Raghunathan Rengaswamy. “Robust and reliable estimation via unscented recursive nonlinear dynamic data reconciliation”. In: *Journal of process control* 16.10 (2006), pp. 1075–1086.

[412] Pramod Vachhani, Raghunathan Rengaswamy, Vikrant Gangwal, and Shankar Narasimhan. “Recursive estimation in constrained nonlinear dynamical systems”. In: *AIChE Journal* 51.3 (2005), pp. 946–959.

[413] Aad W Van der Vaart. *Asymptotic statistics*. Vol. 3. Cambridge university press, 2000.

[414] Tom Van Herpe, Bert Pluymers, Marcelo Espinoza, Greet Van den Berghe, and Bart De Moor. "A minimal model for glycemia control in critically ill patients". In: *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2006, pp. 5432–5435.

[415] Eric Vanden-Eijnden et al. "Fast communications: Numerical techniques for multi-scale dynamical systems with stochastic effects". In: *Communications in Mathematical Sciences* 1.2 (2003). Publisher: International Press of Boston, pp. 385–391.

[416] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[417] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[418] Geert Verbeke and Geert Molenberghs. *Linear mixed models for longitudinal data*. Springer Science & Business Media, 2009.

[419] Michael Vilkhovoy, Ari Pritchard-Bell, Gilles Clermont, and Robert S Parker. "A control-relevant model of subcutaneous insulin absorption". In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 10988–10993.

[420] Pauli Virtanen et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python". en. In: *Nature Methods* 17.3 (Mar. 2020). Number: 3 Publisher: Nature Publishing Group, pp. 261–272. ISSN: 1548-7105. DOI: 10.1038/s41592-019-0686-2. URL: https://www.nature.com/articles/s41592-019-0686-2 (visited on 04/14/2021).

[421] P.R. Vlachas, J. Pathak, B.R. Hunt, T.P. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos. "Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics". en. In: *Neural Networks* 126 (June 2020), pp. 191–217. ISSN: 08936080. DOI: 10.1016/j.neunet.2020.02.016. URL: https://linkinghub.elsevier.com/retrieve/pii/S0893608020300708 (visited on 05/18/2020).

[422] Dingbao Wang, Yuguo Chen, and Ximing Cai. "State and parameter estimation of hydrologic models using the constrained ensemble Kalman filter". In: *Water resources research* 45.11 (2009).

[423] Jack Wang, Aaron Hertzmann, and David J. Fleet. "Gaussian Process Dynamical Models". en. In: *Advances in Neural Information Processing Systems* 18 (2005). URL: https://papers.nips.cc/paper/2005/hash/ccd45007df44dd0f12098f486e7e8a0f-Abstract.html (visited on 04/22/2021).

[424] Jian-Xun Wang, Jin-Long Wu, and Heng Xiao. “Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data”. In: *Physical Review Fluids* 2.3 (2017), p. 034603.

[425] Ke Alexander Wang, Matthew E. Levine, Jiaxin Shi, and Emily B. Fox. “Learning Absorption Rates in Glucose-Insulin Dynamics from Meal Covariates”. In: *NeurIPS 2022 Workshop on Learning from Time Series for Health*. 2023. arXiv: 2304.14300 [cs.LG].

[426] Qian Wang, Nicolo Ripamonti, and Jan S Hesthaven. “Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism”. In: *Journal of Computational Physics* 410 (2020), p. 109402.

[427] Qian Wang, Nicolò Ripamonti, and Jan S Hesthaven. “Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism”. In: *Journal of Computational Physics* 410 (2020). Publisher: Elsevier, p. 109402.

[428] Shuchun Wang, Wei Chen, and Kwok-Leung Tsui. “Bayesian validation of computer models”. In: *Technometrics* 51.4 (2009), pp. 439–451.

[429] Peter A. G. Watson. “Applying machine learning to improve simulations of a chaotic dynamical system using empirical error correction”. In: *arXiv:1904.10904 [nlin, physics:physics, stat]* (Apr. 2019). arXiv: 1904.10904. URL: http://arxiv.org/abs/1904.10904 (visited on 04/30/2019).

[430] Jack Weatheritt and Richard Sandberg. “A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship”. In: *Journal of Computational Physics* 325 (2016), pp. 22–37.

[431] Irving B Weiner and W Edward Craighead. *The Corsini encyclopedia of psychology*. Vol. 4. John Wiley & Sons, 2010.

[432] Alexander Wikner, Jaideep Pathak, Brian Hunt, Michelle Girvan, Troy Arcomano, Istvan Szunyogh, Andrew Pomerance, and Edward Ott. “Combining machine learning with knowledge-based modeling for scalable forecasting and subgrid-scale closure of large, complex, spatiotemporal systems”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.5 (May 2020). Publisher: American Institute of Physics, p. 053111. ISSN: 1054-1500. DOI: 10.1063/5.0005541. URL: https://aip.scitation.org/doi/10.1063/5.0005541 (visited on 06/02/2021).

[433] Darren J Wilkinson. “Stochastic modelling for quantitative description of heterogeneous biological systems”. In: *Nature Reviews Genetics* 10.2 (2009), pp. 122–133.

[434] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. “Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems”. In: *arXiv:2003.04919 [physics, stat]* (July

2021). arXiv: 2003.04919. URL: http://arxiv.org/abs/2003.04919 (visited on 08/04/2021).

[435] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. "Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems". In: *ACM Computing Surveys* (Mar. 2022), p. 3514228. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3514228.

[436] Christopher K Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[437] J.A. Wilson and L.F.M. Zorzetto. "A generalised approach to process state estimation using hybrid artificial neural network/mechanistic models". en. In: *Computers & Chemical Engineering* 21.9 (June 1997), pp. 951–963. ISSN: 00981354. DOI: 10.1016/S0098-1354(96)00336-5. URL: http://linkinghub.elsevier.com/retrieve/pii/S0098135496003365 (visited on 01/30/2019).

[438] Armand Wirgin. "The inverse crime". In: *arXiv:math-ph/0401050* (Jan. 2004). arXiv: math-ph/0401050. URL: http://arxiv.org/abs/math-ph/0401050 (visited on 04/08/2021).

[439] David H. Wolpert. "Stacked generalization". en. In: *Neural Networks* 5.2 (Jan. 1992), pp. 241–259. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(05)80023-1. URL: https://www.sciencedirect.com/science/article/pii/S0893608005800231 (visited on 04/06/2021).

[440] Jin-Long Wu, Heng Xiao, and Eric Paterson. "Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework". In: *Physical Review Fluids* 3.7 (2018), p. 074602.

[441] Sha Wu and Eiko Furutani. "Nonlinear model predictive glycemic control of critically ill patients using online identification of insulin sensitivity". In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2016, pp. 2245–2248.

[442] Heng Xiao, J-L Wu, J-X Wang, Rui Sun, and CJ Roy. "Quantifying and reducing model-form uncertainties in Reynolds-averaged Navier–Stokes simulations: A data-driven, physics-informed Bayesian approach". In: *Journal of Computational Physics* 324 (2016), pp. 115–136.

[443] Jun Yang, Lei Li, Yimeng Shi, and Xiaolei Xie. "An ARIMA model with adaptive orders for predicting blood glucose concentrations and hypoglycemia". In: *IEEE journal of biomedical and health informatics* 23.3 (2018), pp. 1251–1260.

[444] Xiongtan Yang, Biao Huang, and Vinay Prasad. "Inequality constrained parameter estimation using filtering approaches". In: *Chemical Engineering Science* 106 (2014), pp. 211–221.

[445] Zhong Yi Wan, Petr Karnakov, Petros Koumoutsakos, and Themistoklis P. Sapsis. “Bubbles in turbulent flows: Data-driven, kinematic models with history terms”. en. In: *International Journal of Multiphase Flow* 129 (Aug. 2020), p. 103286. ISSN: 0301-9322. DOI: 10.1016/j.ijmultiphaseflow.2020.103286. URL: http://www.sciencedirect.com/science/article/pii/S030193221930758X (visited on 09/07/2020).

[446] Yuan Yin, Vincent Le Guen, Jérémie Dona, Emmanuel de Bézenac, Ibrahim Ayed, Nicolas Thome, and Patrick Gallinari. “Augmenting Physical Models with Deep Networks for Complex Dynamics Forecasting”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.12 (Dec. 2021), p. 124012. ISSN: 1742-5468. DOI: 10.1088/1742-5468/ac3ae5.

[447] Janni Yuval, Paul A O’Gorman, and Chris N Hill. “Use of neural networks for stable, accurate and physically consistent parameterization of subgrid atmospheric processes with good performance at reduced precision”. In: *Geophysical Research Letters* 48.6 (2021), e2020GL091363.

[448] Laure Zanna and Thomas Bolton. “Data-driven equation discovery of ocean mesoscale closures”. In: *Geophysical Research Letters* 47.17 (2020), e2020GL088376.

[449] Laure Zanna and Thomas Bolton. “Deep Learning of Unresolved Turbulent Ocean Processes in Climate Models”. In: *Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science, and Geosciences* (2021), pp. 298–306.

[450] David Zeevi, Tal Korem, Niv Zmora, David Israeli, Daphna Rothschild, Adina Weinberger, Orly Ben-Yacov, Dar Lador, Tali Avnit-Sagi, Maya Lotan-Pompan, et al. “Personalized nutrition by prediction of glycemic responses”. In: *Cell* 163.5 (2015), pp. 1079–1094.

[451] David Zeevi et al. “Personalized Nutrition by Prediction of Glycemic Responses”. In: *Cell* 163.5 (Nov. 2015), pp. 1079–1094. ISSN: 00928674. DOI: 10.1016/j.cell.2015.11.001.

[452] He Zhang, John Harlim, and Xiantao Li. “Error Bounds of the Invariant Statistics in Machine Learning of Ergodic Itó Diffusions”. en. In: *arXiv:2105.10102 [cs, math]* (May 2021). arXiv: 2105.10102. URL: http://arxiv.org/abs/2105.10102 (visited on 06/02/2021).

[453] He Zhang, John Harlim, Xiantao Li, ,Department of Mathematics, The Pennsylvania State University, University Park, PA 16802, USA, and ,Department of Mathematics, Department of Meteorology and Atmospheric Science, Institute for Computational and Data Sciences, The Pennsylvania State University, University Park, PA 16802, USA. “Estimating linear response statistics using orthogonal polynomials: An RKHS formulation”. en. In: *Foundations of Data Science* 2.4 (2020), pp. 443–485. ISSN: 2639-8001. DOI:

10.3934/fods.2020021. URL: http://aimsciences.org//article/doi/10.3934/fods.2020021 (visited on 10/25/2021).

[454] Lan Zhang. "Efficient estimation of stochastic volatility using noisy observations: A multi-scale approach". In: *Bernoulli* 12.6 (2006), pp. 1019–1043.

[455] Lan Zhang, Per A Mykland, and Yacine Ait-Sahalia. "A tale of two time scales: Determining integrated volatility with noisy high-frequency data". In: *Journal of the American Statistical Association* 100.472 (2005), pp. 1394–1411.

[456] Xin-Lei Zhang, Heng Xiao, Xiaodong Luo, and Guowei He. "Ensemble-based learning of turbulence model from indirect observation data". In: *arXiv preprint arXiv:2202.05122* (2022).

[457] Yan Zhang, Tim A Holt, and Natalia Khovanova. "A data driven nonlinear stochastic model for blood glucose dynamics". In: *Computer methods and programs in biomedicine* 125 (2016), pp. 18–25.

[458] Zhongheng Zhang. "A mathematical model for predicting glucose levels in critically-ill patients: the PIGnOLI model". In: *PeerJ* 3 (2015), e1005.

[459] Yaomin Zhao, Harshal D Akolekar, Jack Weatheritt, Vittorio Michelassi, and Richard D Sandberg. "RANS turbulence model development using CFD-driven machine learning". In: *Journal of Computational Physics* 411 (2020), p. 109413.

[460] Xu-Hui Zhou, Jiequn Han, and Heng Xiao. "Learning nonlocal constitutive models with neural networks". In: *Computer Methods in Applied Mechanics and Engineering* 384 (2021), p. 113927.

[461] Jian Zhu and Masafumi Kamachi. "An adaptive variational method for data assimilation with imperfect models". In: *Tellus A: Dynamic Meteorology and Oceanography* 52.3 (Jan. 2000). Publisher: Taylor & Francis _eprint: https://doi.org/10.3402/tellusa.v52i3.12265, pp. 265–279. ISSN: null. DOI: 10.3402/tellusa.v52i3.12265. URL: https://doi.org/10.3402/tellusa.v52i3.12265 (visited on 04/05/2021).

[462] Yuanran Zhu, Jason M Dominy, and Daniele Venturi. "On the estimation of the Mori-Zwanzig memory integral". In: *Journal of Mathematical Physics* 59.10 (2018). Publisher: AIP Publishing LLC, p. 103501.

[463] Jiawei Zhuang, Dmitrii Kochkov, Yohai Bar-Sinai, Michael P. Brenner, and Stephan Hoyer. "Learned discretizations for passive scalar advection in a two-dimensional turbulent flow". In: *Phys. Rev. Fluids* 6 (6 June 2021), p. 064605.

[464] Raed Abu Zitar and Abdulkareem Al-Jabali. "Towards neural network model for insulin/glucose in diabetics-II". In: *Informatica* 29.2 (2005).

[465] Robert Zwanzig. *Nonequilibrium statistical mechanics*. Oxford university press, 2001.