Learning Physics Informed Neural ODEs with Partial Measurements

Paul Ghanem[1], **Ahmet Demirkaya**[1], **Tales Imbiriba**[2], **Alireza Ramezani**[1], **Zachary Danziger**[3], **Deniz Erdogmus**[1]

[1]Northeastern University, Boston Massachusetts
[2]University of Massachusetts, Boston Massachusetts
[3]Emory University, Atlanta Georgia
{ghanem.p, demirkaya.a}@northeastern.edu, tales.imbiriba@umb.edu, a.ramezani@northeastern.edu, zdanzige@fiu.edu, d.erdogmus@northeastern.edu

Abstract

Learning dynamics governing physical and spatiotemporal processes is a challenging problem, especially in scenarios where states are partially measured. In this work, we tackle the problem of learning dynamics governing these systems when parts of the system's states are not measured, specifically when the dynamics generating the non-measured states are unknown. Inspired by state estimation theory and Physics Informed Neural ODEs, we present a sequential optimization framework in which dynamics governing unmeasured processes can be learned. We demonstrate the performance of the proposed approach leveraging numerical simulations and a real dataset extracted from an electro-mechanical positioning system. We show how the underlying equations fit into our formalism and demonstrate the improved performance of the proposed method when compared with baselines.

1 Introduction

Ordinary differential equations (ODEs) are used to describe the state evolution of many complex physical systems in engineering, biology, and other fields of natural sciences. Traditionally, first-principle notions are leveraged in designing ODEs as a form to impose physical meaning and interpretability (Psichogios and Ungar 1992) of latent states. A major issue, however, is the inherent complexity of real-world problems for which even carefully designed ODE systems cannot account for all aspects of the true underlying physical phenomenon (Karniadakis et al. 2021). Moreover, we often require prediction of systems whose dynamics are not fully understood or are partially unknown.

In this context, Neural ODEs (NODEs) (Chen et al. 2018) emerged as a powerful tool for learning complex correlations directly from the data, where residual neural networks (NNs) are used to parameterize the hidden ODEs' states. Extensions of NODE were developed to improve learning speed (Xia et al. 2021; Massaroli et al. 2021) and learning longtime dependencies in irregularly sampled time series(Xia et al. 2021). A major challenge in learning NODEs arises when latent states of interest contribute indirectly to the measurements. This is the case when an unmeasured state influences a measured state. In this scenario, NODE's standard solutions, which are optimized using the adjoint method (Boltyanskiy et al. 1962), are compromised. Furthermore, NODE systems may have infinitely many solutions since parameters and unmeasured states are estimated jointly. As a consequence, even when the model is capable of fitting the data, unmeasured states cannot be accurately inferred without constraining the solution space (Demirkaya et al. 2021). To constrain the solution space, hybrid and physics informed neural ODEs were developed to incorporate physical knowledge of the system being learned when available(Sholokhov et al. 2023; O'Leary, Paulson, and Mesbah 2022). Despite their recent success in neural ODEs and neural networks in general, these methods lack the ability to learn dynamical systems under the partial measurements scenario when dynamics generating unmeasured states are unknown. Moreover, hybrid and physics informed strategies were leveraged to obtain estimations of missing states under partial measurements scenario (Imbiriba et al. 2022; Demirkaya et al. 2021; Ghanem et al. 2021). Despite the lack of a clear formalization, in these works the authors were imposing some kind of identifiability among states by adding known parts of the dynamics, resulting in hybrid first-principle data-driven models. Nevertheless, these works focus on state estimation using data-driven components to improve or augment existing dynamics but fail to learn global models and do not scale well for large models.

In this paper, we propose a sequential alternating second order optimization approach that, unlike previous approaches, solves at each time step an alternating optimization problem for learning system dynamics under partially measured states, when states are identifiable. The approach focuses on learning unknown dynamics from data where the state related to the unknown dynamics is unmeasured. Since the unobserved dynamics are unknown, we assume it is described by parametric models such as NNs. We propose aiding a model's training with the knowledge of the physics regarding the measured states using a physics-informed loss term. The motivation for using sequential optimization is to make second order optimization feasible, while the motivation for using alternating optimization is to enable usage of hidden states to learn model parameters, which is central for optimization success in partial observation scenarios. The proposed approach combines the two optimization frame-

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

works, sequential and alternating, in one approach. The benefit of the proposed approach is twofold: (1) reduce the need for accurate initial conditions during training; (2) enables usage of hidden states estimates to learn model parameters instead of simultaneous estimation of states and parameters, making second-order optimization methods feasible under partial measurement scenario. Furthermore, the proposed approach exploits the identifiable property of states by designing an alternating optimization strategy with respect to states and parameters. The result is an interconnected optimization procedure, where at each step model parameters and data are used to estimate latent states, and corrected latent states are used to update the model parameters in the current optimization step. Moreover, we define identifiable latent variables and test our proposed approach in hybrid scenarios where NNs replace parts of the ODE systems such that the identifiability of latent variables is kept. Finally, as a side effect of the recursive paradigm adopted the proposed strategy can assimilate data and estimate initial conditions by leveraging its sequential state estimation framework over past data.

2 Related Work

Partial Measurements: In the context of data-driven ODE designs, most learning frameworks assume that all states are measured in the sense that they are directly measured. This assumption does not reflect many real-world scenarios where a subset of the states are unmeasured. GP-SSM is a well-established approach used for dynamic systems identification (McHutchon et al. 2015; Ialongo et al. 2019). GP-SSM can be adapted by introducing a recognition model that maps outputs to latent states to solve the problem of partial measurements (Eleftheriadis et al. 2017). Nevertheless, these methods do not scale well with large datasets and are limited to small trajectories(Doerr et al. 2018). Indeed, (Doerr et al. 2018) minimizes this problem by using stochastic gradient ELBO optimization on minibatches. However, GP-SSM-based methods avoid learning the vector field describing the latent states and instead directly learn a mapping from a history of past inputs and measurements to the next measurement.

Similar approaches to recognition models have been used for Bayesian extensions of Neural Ordinary Differential Equations (NODEs). In these extensions, the NODE describes the dynamics of latent states, while the distribution of the initial latent variable given the measurements are approximated by encoder and decoder networks (Yildiz, Heinonen, and Lahdesmaki 2019; Norcliffe et al. 2021). The encoder network, which links measurements to latent states by a deterministic mapping or by approximating the conditional distribution, can also be a Recurrent Neural Network (RNN) (Rubanova, Chen, and Duvenaud 2019; Kim et al. 2021; De Brouwer et al. 2019), or an autoencoder (Bakarji et al. 2023). Despite focusing on mapping measurements to latent states with neural networks and autoencoders, these works were not demonstrated to learn parameterized models under partial measurements. Moreover, this parameterized line of work of mapping measurement to latent states suffers from unidentifiability problem since several latent inputs could lead to the same measurement. Recently, sparse approaches such as (Bakarji et al. 2022) merged encoder networks to identify a parsimonious transformation of the hidden dynamics of partially measured latent states. Moreover, Nonlinear Observers and recognition models were combined with NODEs to learn dynamic model parameters from partial measurements while enforcing physical knowledge in the latent space (Buisson-Fenet et al. 2022). Differently from the aforementioned methods, in this work, we propose a recursive alternating approach that uses alternating Newton updates to optimize a quadratic cost function with respect to states and model parameters.

Second order Newton method: Despite the efficiency and popularity of many stochastic gradient descent methods (Robbins and Monro 1951; Duchi, Hazan, and Singer 2011; Hinton, Srivastava, and Swersky 2012; Kingma and Ba 2014) for optimizing NNs, great efforts have been devoted to exploiting second-order Newton methods where Hessian information is used, providing faster convergence (Martens and Grosse 2015; Botev, Ritter, and Barber 2017; Gower, Goldfarb, and Richtárik 2016; Mokhtari and Ribeiro 2014). When training neural networks, computing the inverse of the Hessian matrix can be extremely expensive (Goldfarb, Ren, and Bahamou 2020) or even intractable. To mitigate this issue, Quasi-Newton methods have been proposed to approximate the Hessian pre-conditioner matrix such as Shampoo algorithm (Gupta, Koren, and Singer 2018), which was extended in (Anil et al. 2020) to simplify blocks of the Hessian, and in (Gupta, Koren, and Singer 2018) to be used in variational inference second-order approaches (Peirson et al. 2022). Similarly, works in (Goldfarb, Ren, and Bahamou 2020; Byrd et al. 2016) focused on developing stochastic quasi-Newton algorithms for problems with large amounts of data. It was shown that recursive the extended Kalman filter can be viewed as Gauss-Newton method (Bell 1994; Bertsekas 1996). Moreover, Newton's method was used to derive recursive estimators for prediction and smoothing (Humpherys, Redd, and West 2012). In this paper, we develop a recursive Newton method that mitigates the problem of partial measurements of latent states.

3 Model and Background

In this section, we describe our modeling assumptions, discuss the identifiability of latent states, and present the time evolution of the resulting generative model.

Model

In this work, we focus on dynamical models characterized by a set of ordinary differential equations describing the time evolution of system states $x(t)$ and system parameters $\theta(t)$, and a measurement equation outputting measurements $y(t) \in \mathcal{Y} \subset \mathbb{R}^{d_y}$ of a subset of these states. These models can be described as follows (Särkkä and Svensson 2023):

$$\begin{aligned} \dot{\theta}(t) &= \tilde{\nu}(t) \\ \dot{x}(t) &= f(x(t), u(t), \overbrace{a(x(t), \theta(t))}^{\text{Hidden physics}}) + \tilde{\epsilon}(t) \\ y(t) &= h(x(t)) + \zeta(t) \end{aligned} \tag{1}$$

where $x(t) \in \mathcal{X} \subset \mathbb{R}^{d_x}$ are systems states and $\theta(t) \in \mathcal{P} \subset \mathbb{R}^{d_\theta}$ are system parameters. $a : \mathcal{X} \times \mathcal{P}$ represents a system of hidden ODE parameterized by $\theta(t)$ that needs to be learned without measurement available. $f : \mathcal{X} \times \mathcal{P} \times \mathcal{U}$ represents a system of ODEs parameterized by $\theta(t)$, where each equation describes the time dynamics of an individual component of a dynamical system. $h : \mathcal{X} \to \mathcal{Y}$ represents the measurement function. $u(t) \in \mathcal{U} \subset \mathbb{R}^{d_u}$ is a vector of external inputs, $\tilde{\nu}(t) \sim \mathcal{N}(0, \tilde{Q}_\theta), \tilde{\epsilon}(t) \sim \mathcal{N}(0, \tilde{Q}_x)$, and $\zeta(t) \sim \mathcal{N}(0, R_y)$, are zero mean white noise independent of $x(t), \theta(t)$ and $y(t)$. The subscript t indicates vectors that vary through time.

The partial measurement problem: Ideally, states $x(t)$ would be directly measured, and thus appear as an element in $y(t)$. In practice, some of these states could influence $y(t)$ only indirectly by acting on other measurable states, where $d_y < d_x$. That is when classical training fails. In this work, we are interested in learning the unknown dynamics $a(x(t), \theta(t))$ governing unmeasured states $x_h(t)$, where

$$\dot{x}_h(t) = a(x(t), \theta(t)) + \tilde{\epsilon}_h(t) \tag{2}$$

where $x_h(t) \subset x(t)$ and $\tilde{\epsilon}_h(t) \subset \tilde{\epsilon}(t)$.This scenario poses further challenges over the estimation process since the recovery of latent states can be compromised.

Identifiability of latent states: The task of recovering latent states $x(t)$ from a sequence of measurements and inputs $\mathcal{D}_N \triangleq \{u(0), y(0), \dots, u(N-1), y(N-1)\}$ depends on the relationship between measurements and latent states. This problem gets even more complicated when model parameters need to be simultaneously estimated. Recent works define different forms of identifiability to analyze scenarios where latent states and parameters can be properly estimated (Wieland et al. 2021). Specifically, they define identifiability of latent variables $x(t)$ as follows:

Definition 1 (State Identifiability) *We say that latent variable $x(t_a)$ is identifiable given a parameter value $\hat{\theta}(t)$ and a measurement sequence $y(t) \in \mathcal{Y} \subset \mathbb{R}^{d_y}$ if (Wang, Blei, and Cunningham 2021; Wieland et al. 2021)*

$$x(t_a) \neq x(t_b) \implies h(x(t_a)) \neq h(x(t_b)) \,. \tag{3}$$

Although it is extremely difficult to provide formal guarantees, it makes sense that if for a given parameter $\hat{\theta}(t)$, $h(x(t_a)) = h(x(t_b))$, then obtaining an estimator for true state $x(t)$ becomes extremely challenging if not unfeasible. Since the proposed approach in this paper relies on estimating unmeasured states to learn model parameters, state identifiability is preferred.

To enforce latent variable identifiability, it is sufficient to ensure that the measurement function h is an injective function (Wang, Blei, and Cunningham 2021) for all θ. Nevertheless, constructing an injective measurement function requires that $d_y \geq d_x$ (Wang, Blei, and Cunningham 2021), which is not feasible when dealing with the partial measurement problem where $d_y < d_x$. Moreover, for models with a high number of connected parameters, such as neural networks, enforcing identifiabilities can be challenging (Wieland et al. 2021) and latent identifiability as defined in Definition 1 is not always guaranteed, especially when the number of measured states is less than the number of latent states. Note that a latent variable may be identifiable in a model given one dataset but not another, and at one θ but not another (Wang, Blei, and Cunningham 2021). However, one could argue that one way to impose state identifiability is to re-parameterize the model (Wieland et al. 2021) and incorporate prior knowledge regarding the relationship of states, focusing on achieving the properties stated in Definition 1.

Discrete Generative model

In the continuous model presented in (1),a continuous-time description for the system states and parameters is assumed even though the measurements are recorded at discrete time points. Moreover, a function $a(x(t), \theta(t))$ was defined to describe the NODE governing the dynamics of the unmeasured states, and $u(t)$ described the external control inputs. In what follows, we will omit to use a and $u(t)$ for notation simplicity, where $f(x(t), u(t), a(x(t), \theta(t))$ will be denoted by $f(x(t), \theta(t))$. The discretization of states $x(t)$ and parameters $\theta(t)$ can therefore be expressed as time integration of (1) using Euler-Maruyama method (Särkkä and Svensson 2023) with uniform time step $\Delta_t = t_i - t_{i-1}$:

$$\begin{aligned} x(t_i) &= x(t_{i-1}) + \int_{t_{i-1}}^{t_i} f(x(t), \theta(t))dt + \int_{t_{i-1}}^{t_i} \tilde{\epsilon}(t)dt \\ x(t_i) &= x(t_{i-1}) + \Delta_t f(x(t_{i-1}), \theta(t_{i-1})) + \Delta_t \tilde{\epsilon}(t_{i-1}) \end{aligned} \tag{4}$$

Hence we define the following equation:

$$f_o(x(t_{i-1}), \theta(t_{i-1})) = x(t_{i-1}) + \Delta_t f(x(t_{i-1}), \theta(t_{i-1})) \tag{5}$$

In a similar fashion of states $x(t)$, we discretize the parameters $\theta(t)$ and define the following equation:

$$\theta(t_i)) = \theta(t_{i-1}) + \int_{t_{i-1}}^{t_i} \tilde{\nu}(t)dt = \theta(t_{i-1}) + \nu(t) \tag{6}$$

Based on the continuous model presented in (1) and state discretization presented in (4, 5,6), we present the discrete time evolution of the system states and parameters by the following discrete generative model:

$$\begin{aligned} \theta(t_i) &= \theta(t_{i-1}) + \nu(t) \\ x(t_i) &= f_o(x(t_{i-1}), \theta(t_{i-1})) + \epsilon(t) \\ y(t_i) &= h(x(t_i)) + \zeta(t) \,. \end{aligned} \tag{7}$$

where $\nu(t) \sim \mathcal{N}(0, Q_\theta), \epsilon(t) \sim \mathcal{N}(0, Q_x)$, and $\zeta(t) \sim \mathcal{N}(0, R_y)$, with $Q_\theta = \Delta_t \tilde{Q}_\theta$ and $Q_x = \Delta_t \tilde{Q}_x$.

4 Method

The proposed approach finds the model parameters $\theta(t)$ of hidden Neural ordinary differential equation $a(x, \theta)$ describing $x_h(t) \in x(t)$ and latent states $x(t)$ of dynamical system given a dataset $\mathcal{D} \triangleq \{u(t_0), y(t_0), \dots, u(t_{N-1}), y(t_{N-1})\}$ of discrete measurements and control inputs when $x(t)$ is partially measured, that is $x_h(t)$ is unmeasured. We formulate the problem of estimating $x(t)$ and $\theta(t)$ as an optimization problem that is similar to (Humpherys, Redd, and

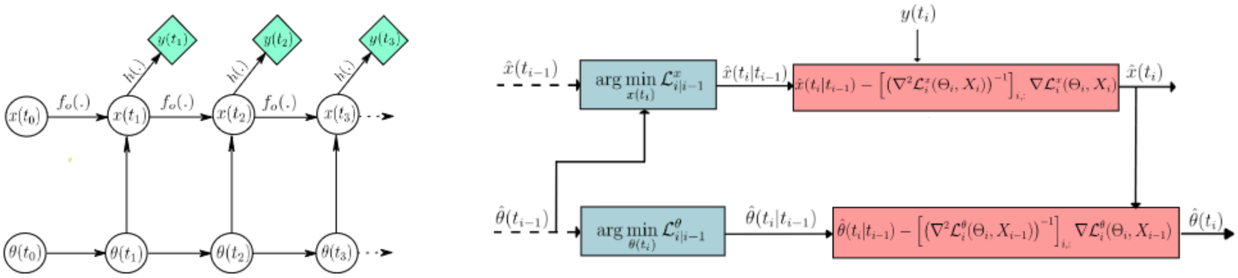
$y(t_1)$
$y(t_2)$
$y(t_3)$
$h(.)$
$h(.)$
$h(.)$
$x(t_0)$
$f_o(.)$
$x(t_1)$
$f_o(.)$
$x(t_2)$
$f_o(.)$
$x(t_3)$
$\theta(t_0)$
$\theta(t_1)$
$\theta(t_2)$
$\theta(t_3)$
$y(t_i)$
$\hat{x}(t_{i-1})$
$\arg\min_{x(t_i)} \mathcal{L}^x_{i|i-1}$
$\hat{x}(t_i|t_{i-1})$
$\hat{x}(t_i|t_{i-1}) - \left[\left(\nabla^2 \mathcal{L}^x_i(\Theta_i, X_i)\right)^{-1}\right]_{i,:} \nabla \mathcal{L}^x_i(\Theta_i, X_i)$
$\hat{x}(t_i)$
$\hat{\theta}(t_{i-1})$
$\arg\min_{\theta(t_i)} \mathcal{L}^\theta_{i|i-1}$
$\hat{\theta}(t_i|t_{i-1})$
$\hat{\theta}(t_i|t_{i-1}) - \left[\left(\nabla^2 \mathcal{L}^\theta_i(\Theta_i, X_{i-1})\right)^{-1}\right]_{i,:} \nabla \mathcal{L}^\theta_i(\Theta_i, X_{i-1})$
$\hat{\theta}(t_i)$

Figure 1: The generative model (left panel), and one step of the proposed optimization strategy (right panel).

West 2012), where we optimize a cost function $\mathcal{L}$ given a probabilistic discrete model (7), exploiting the link between the second-order Newton's method and the Kalman filter. The cost function $\mathcal{L}$ is updated and solved sequentially to find latent states $x(t)$ and model parameters $\theta(t)$ in a unified framework. Our approach assumes state identifiability which implies that latent states $x(t)$ are recoverable from measurements $y(t)$. In this context, we break the optimization steps into two concerning optimization with respect to $x(t)$ and $\theta(t)$.

Sequential Newton Derivation

We denote by $\Theta_N = [\theta(t_0), \dots, \theta(t_N)]$ and $X_N = [x(t_0), \dots, x(t_N)]$ to be the set of latent states sampled at $t_0, t_1, \dots, t_N$. To train the model, we optimize (Θ_N, X_N) to minimize a quadratic cost function starting from initial $\{x(t_0), \theta(t_0)\}$ using a collection of combined measurement and input sequences $\mathcal{D}$. A physics informed loss term is employed in our cost function to help identify hidden states x_h, where the cost function is defined as:

$$\mathcal{L}_N(\Theta_N, X_N) = \frac{1}{2}\sum_{i=1}^{N} \overbrace{\|x(t_i) - f_o(x(t_{i-1}), \theta(t_{i-1}))\|^2_{Q_x^{-1}}}^{\text{Physics Informed loss}} + \underbrace{\|y(t_i) - h(x(t_i))\|^2_{R_y^{-1}}}_{\text{Data driven loss}} + \underbrace{\|\theta(t_i) - \theta(t_{i-1})\|^2_{Q_\theta^{-1}}}_{\text{Regularization term}} . \tag{8}$$

where Q_x, R_y and Q_θ are known positive definite matrices corresponding to latent states, measurement and parameter uncertainty respectively , and $\|a - b\|^2_{A^{-1}} = (a - b)^T A^{-1}(a - b)$. As the Hessian's inverse is in general intractable, finding optimal solution (Θ^*_N, X^*_N) using the second order Newton method over the whole data set of size N is unfeasible. For this reason, we resort to a sequential strategy by introducing a modified quadratic function $\mathcal{L}_i(\Theta_i, X_i)$. Let us re-write the cost function at time t_i as:

$$\begin{aligned}\mathcal{L}_i(\Theta_i, X_i) &= \mathcal{L}_{i-1}(\Theta_{i-1}, X_{i-1}) \\ &+ \frac{1}{2}\|x(t_i) - f_o(x(t_{i-1}), \theta(t_{i-1}))\|^2_{Q_x^{-1}} \\ &+ \frac{1}{2}\|y(t_i) - h(x(t_i))\|^2_{R_y^{-1}} + \frac{1}{2}\|\theta(t_i) - \theta(t_{i-1})\|^2_{Q_\theta^{-1}}\end{aligned} \tag{9}$$

where $\mathcal{L}_{i-1}(\Theta_{i-1}, X_{i-1})$ and $\mathcal{L}_i(\Theta_i, X_i)$ are the cost functions at times t_{i-1} and t_i, respectively; $\Theta_i = [\theta(t_0), \dots, \theta(t_i)]$ and $X_i = [x(t_0), \dots, x(t_i)]$. In the sequential optimization paradigm, Θ_{i-1} and X_{i-1} are assumed known and at the i-th optimization step is performed only with respect to $\{\theta(t_i), x(t_i)\}$. When $\{\theta(t_i), x(t_i)\}$ are determined jointly such as in (Humpherys, Redd, and West 2012), the optimization process will suffer from vanishing gradients under partial measurements, see Appendix B. However, if $x(t_i)$ is identifiable, we can circumvent the vanishing gradient problem by first optimizing with respect to $x(t_i)$ and then $\theta(t_i)$. To improve identifiability of latent states $x(t)$, we employ a physics informed term in the cost function described in 8 and combine it with the alternating optimization approach proposed below, that optimizes $x(t)$ and $\theta(t)$ separately. This will allow us to circumvent the partial observability problem and enable the use of an estimate of the unmeasured state in training. To do so, we break the optimization function (9) into four alternating optimization procedures aiming at finding $\hat{x}(t_i)$ and then finding $\hat{\theta}(t_i)$ that minimizes (9) given $\hat{x}(t_i)$.

Let us begin by defining two intermediate optimization functions $\mathcal{L}^x_{i|i-1}$ and $\mathcal{L}^\theta_{i|i-1}$ in (10) and (11) respectively as follows:

$$\mathcal{L}^x_{i|i-1}(\Theta_i, X_i) = \mathcal{L}_{i-1}(\Theta_{i-1}, X_{i-1}) + \frac{1}{2}\|x(t_i) - f_o(x(t_{i-1}), \theta(t_{i-1}))\|^2_{Q_x^{-1}} + \frac{1}{2}\|\theta(t_i) - \theta(t_{i-1})\|^2_{Q_\theta^{-1}} \tag{10}$$

$$\mathcal{L}^\theta_{i|i-1}(\Theta_i, X_{i-1}) = \mathcal{L}_{i-1}(\Theta_{i-1}, X_{i-1}) + \frac{1}{2}\|\theta(t_i) - \theta(t_{i-1})\|^2_{Q_\theta^{-1}} \tag{11}$$

We proceed by optimizing (10) for $x(t_i)$ and (11) for $\theta(t_i)$, yielding the respective solutions below:

$$\begin{aligned}\hat{\theta}(t_i|t_{i-1}) &= \hat{\theta}(t_{i-1}) \\ \hat{x}(t_i|t_{i-1}) &= f_o(\hat{x}(t_{i-1}), \hat{\theta}(t_{i-1})) \, .\end{aligned} \tag{12}$$

Next, we define the two optimization functions responsible for the update steps for states and parameters. Specifically, we define $\mathcal{L}^x_i$ as:

$$\mathcal{L}^x_i(\Theta_i, X_i) = \mathcal{L}^x_{i|i-1}(\Theta_i, X_i) + \|y(t_i) - h(x(t_i))\|^2_{R_y^{-1}} \tag{13}$$

to be optimized with respect to $x(t_i)$ by minimizing $\mathcal{L}^x_i$ given intermediate values of equation (12) where:

$$\hat{x}(t_i) = \hat{x}(t_i|t_{i-1}) - \left[\left(\nabla^2 \mathcal{L}^x_i(\Theta_i, X_i)\right)^{-1}\right]_{i,:} \nabla \mathcal{L}^x_i(\Theta_i, X_i) \, .$$

The solution to the above problem is given by given by (15). Equivalently, we define the update optimization function $\mathcal{L}_i^\theta$ as:

$$\mathcal{L}_i^\theta(\Theta_i, X_i) = \mathcal{L}_{i|i-1}^\theta(\Theta_i, X_{i-1}) \tag{14}$$
$$+ \|x(t_i) - f_o(x(t_{i-1}), \theta(t_{i-1}))\|_{Q_x^{-1}}^2 + \|y(t_i) - h(x(t_i))\|_{R_y^{-1}}^2$$

to be optimized with respect to $\theta(t_i)$ by minimizing $\mathcal{L}_i^\theta$ given intermediate values of equation (12) and (15) as follows:

$$\hat{\theta}(t_i) = \hat{\theta}(t_i|t_{i-1}) - \left[\left(\nabla^2 \mathcal{L}_i^\theta(\Theta_i, X_{i-1})\right)^{-1}\right]_{i,:} \nabla \mathcal{L}_i^\theta(\Theta_i, X_{i-1})$$

The resulting optimal variable $\hat{\theta}(t_i)$ is given by (16). The procedure is repeated until $t_i = t_N$. We present our main result in the following theorem:

Theorem 1 *Given $\hat{\theta}(t_{i-1}) \in \hat{\Theta}_{i-1}$ and $\hat{x}(t_{i-1}) \in \hat{X}_{i-1}$, and known $P_{\theta_{i-1}} \in R^{d_\theta \times d_\theta}$ and $P_{x_{i-1}} \in R^{d_x \times d_x}$, the recursive equations for computing $\hat{x}(t_i)$ and $\hat{\theta}(t_i)$ that minimize (9) are given by the following:*

$$\hat{x}(t_i) = f_o(\hat{x}(t_{i-1}), \hat{\theta}(t_{i-1})) - \tag{15}$$
$$P_{x_i}^- H_i^T \Big(H_i P_{x_i}^- H_i^T + R_y\Big)^{-1} \Big[h\Big(f_o(\hat{x}(t_{i-1}), \hat{\theta}(t_{i-1}))\Big) - y(t_i)\Big]$$
$$\hat{\theta}(t_i) = \hat{\theta}(t_{i-1}) - P_{\theta_i}^- F_{\theta_{i-1}}^T \Big[f_o(\hat{x}(t_{i-1}), \hat{\theta}(t_{i-1})) - \hat{x}(t_i)\Big] \tag{16}$$

with $P_{\theta_i}^-$, $P_{x_i}^-$ being intermediate matrices and P_{θ_i} and P_{x_i} being the lower right blocks of $(\nabla^2 \mathcal{L}_i^\theta)^{-1}$ and $(\nabla^2 \mathcal{L}_i^x)^{-1}$ respectively:

$$P_{\theta_i}^- = P_{\theta_{i-1}} - P_{\theta_{i-1}} F_{\theta_{i-1}}^T \Big(Q_x + F_{\theta_{i-1}} P_{\theta_{i-1}} F_{\theta_{i-1}}^T\Big) F_{\theta_{i-1}} P_{\theta_{i-1}}$$
$$P_{x_i}^- = F_{x_{i-1}} P_{x_{i-1}} F_{x_{i-1}} + Q_x \tag{17}$$
$$P_{x_i} = P_{x_i}^- [I + H_i \left(R_y - H_i P_{x_i}^- H_i^T\right) H_i P_{x_i}^-]$$
$$P_{\theta_i} = Q_\theta + P_{\theta_i}^-$$

with $H_i, F_{x_{i-1}}$, and $F_{\theta_{i-1}}$ being the jacobians of the vector fields h and f_o at $\hat{x}(t_i|t_{i-1}), \hat{x}(t_{i-1})$ and $\hat{\theta}(t_{i-1})$: $H_i = \frac{\partial h(\hat{x}(t_i|t_{i-1}))}{\partial \hat{x}(t_i|t_{i-1})}$, $F_{x_{i-1}} = \frac{\partial f_o(\hat{x}(t_{i-1}), \hat{\theta}(t_{i-1}))}{\partial \hat{x}(t_{i-1})}$ *and* $F_{\theta_{i-1}} = \frac{\partial f_o(\hat{x}(t_{i-1}), \hat{\theta}(t_{i-1}))}{\partial \hat{\theta}(t_{i-1})}$

The proof of Theorem 1 is provided in Appendix A. As a consequence of Theorem (1), $\hat{x}(t_i)$ is computed according to (15) using $\hat{\theta}(t_{i-1})$. $\hat{\theta}(t_i)$ is computed afterwards according to (16) using $\hat{x}(t_i)$ that was previously found in (15). This alternating procedure between $x(t_i)$ and $\theta(t_i)$ is explained in the right panel of Figure 1, which depicts the four alternate optimization steps performed for each iteration t_i. The computational complexity of the proposed approach is detailed in Appendix F. An epoch has a complexity of $\mathcal{O}(N(d_x^3 + 2d_\theta^2 d_x + 2d_\theta d_x^2))$. Under the assumption that $d_\theta \gg d_x$ the complexity becomes $\mathcal{O}(N(2d_\theta^2 d_x + 2d_\theta d_x^2))$. During testing, however, the complexity becomes $\mathcal{O}(d_\theta)$ per step if integrating the learned mean vector field.

5 Experiments

The performance of the proposed approach is assessed in comparison to state-of-the-art model learning methods on several challenging nonlinear simulations and real-world datasets. We employed five different dynamical models to demonstrate the effectiveness of the proposed approach. For each dynamical model, we assumed that we don't have parts of the governing dynamics available, and replaced them with a neural network. Euler integrator is used as the ODE solver for efficiency and fast computation speed.

As benchmark methods, we considered five other well-established techniques for dynamical machine learning, namely NODE (Chen et al. 2018), NODE-LSTM (Chen et al. 2018), SPINODE (O'Leary, Paulson, and Mesbah 2022), RM (Buisson-Fenet et al. 2022) and PR-SSM (Doerr et al. 2018). We denote that NODE-LSTM is a NODE with an LSTM network, and SPINODE is a Stochastic Physics Informed NODE. Currently, no code is available for the model learning frameworks presented in (Eleftheriadis et al. 2017). Moreover, the available code related to the works in (McHutchon et al. 2015; Ialongo et al. 2019) could be modified to account for the partial measurement scenario. However, these algorithms become computationally unfeasible for medium and large datasets (Doerr et al. 2018). For that reason, we were not able to benchmark against these approaches. We emphasize that modifying the abovementioned methods to either account for the ODE structure or make them computationally tractable is out of the scope of this paper. This also applies to the PRSSM method. Nevertheless, for the sake of providing comparative results, we still include results using PR-SSM which is computationally more efficient than other Gaussian process-based models but does not account for the ODE structure.

The benchmark results are summarized in Table 1 which represents normalized Root Mean Square Error (nRMSE) values for each model and method. In Figs. 2-5 we compare the benchmark methods, and our proposed method. All results were obtained with learned mean vector field integrated over time. Each subfigure represents the dynamics of a single state and contains ODE solutions for each method. We computed nRMSE using $\text{nRMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^n (x(t_i) - \hat{x}(t_i))^2} / \big[\max(x(t)) - \min(x(t))\big]$, where $\hat{x}(t_i)$ and $x(t_i)$ are the estimated and true states at time t_i, respectively, and n is the number of data points. Moreover, the learning curves for the proposed approach on each example are presented in Appendix E.

Hodgkin-Huxley Neuron Model

The Hodgkin-Huxley (HH) Neuron Model (Hodgkin and Huxley 1952) is an ODE system that describes the membrane dynamics of action potentials in neurons, which are electrical signals used by neurons to communicate with each other. The model has four states: V_m is the membrane potential, n_{gate}, m_{gate}, and h_{gate} are gating variables controlling the membrane's ionic permeability. The equations governing the ODE system are provided in (54)-(57) of the Appendix C. We train our recursive model with the assumption that Eq. (57) governing dynamics of h_{gate} is unknown and its corresponding state is not measured, i.e., $y(t_i) = (V_m(t_i), n_{gate}(t_i), m_{gate}(t_i))$. We replace the dynamics describing $\dot{h}_{gate}(t)$ by a neural network consisting of

Methods	HH model	Yeast Glyco.	Cart-pole	Harmonic Osc.	EMPS
RM (Buisson-Fenet et al. 2022)	$2.39 \cdot 10^{-1}$	$6.30 \cdot 10^{-1}$	$1.06 \cdot 10^{0}$	$2.36 \cdot 10^{-2}$	$6.20 \cdot 10^{-1}$
PR-SSM (Doerr et al. 2018)	$4.05 \cdot 10^{-1}$	$1.59 \cdot 10^{0}$	$1.52 \cdot 10^{0}$	$1.21 \cdot 10^{0}$	$4.05 \cdot 10^{1}$
SPINODE (O'Leary, Paulson, and Mesbah 2022)	$7.68 \cdot 10^{-1}$	$4.98 \cdot 10^{-2}$	$3.01 \cdot 10^{0}$	$4.34 \cdot 10^{-1}$	$4.30 \cdot 10^{5}$
NODE (Chen et al. 2018)	$7.03 \cdot 10^{1}$	$3.74 \cdot 10^{-1}$	$2.84 \cdot 10^{-1}$	$4.65 \cdot 10^{-1}$	$1.65 \cdot 10^{0}$
NODE-LSTM (Chen et al. 2018)	$3.87 \cdot 10^{1}$	$3.09 \cdot 10^{-1}$	$2.90 \cdot 10^{-1}$	$4.60 \cdot 10^{-1}$	$3.45 \cdot 10^{0}$
Proposed Approach	$\mathbf{1.54 \cdot 10^{-1}}$	$\mathbf{3.39 \cdot 10^{-2}}$	$\mathbf{9.41 \cdot 10^{-3}}$	$\mathbf{5.08 \cdot 10^{-3}}$	$\mathbf{9.50 \cdot 10^{-2}}$

Table 1: Comparison of nRMSE values for different dynamical models and methods.

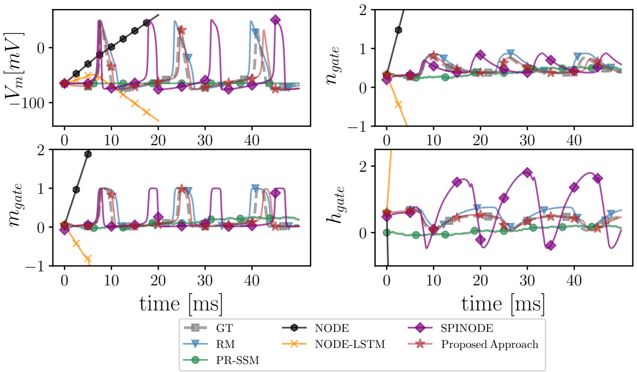

Figure 2: Learned state trajectories of HH model after training with RM, PR-SSM, NODE, NODE-LSTM, SPINODE methods and our proposed approach. Results are compared to ground truth ODE system trajectory labeled as GT. The proposed approach is capable of discerning the true trajectory for the unmeasured state h_{gate}.

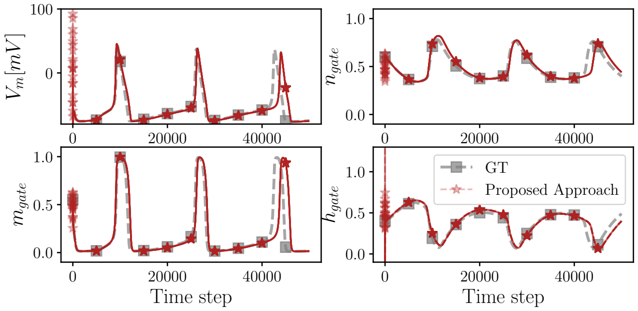

Figure 3: The proposed approach's results for unknown initial conditions. Initial conditions $\hat{x}(t_{100})$ were learned using the first 100 samples.

three feed-forward layers for all the benchmark methods except NODE-LSTM where we used three LSTM layers . The first layer is a 20 units layer followed by an Exponential Linear Unit (ELU) activation function, the second layer is also a 20 unit layer followed by a tanh activation function. The last layer consists of 10 units with a sigmoid activation function. We generate the dataset by applying a constant control input $u(t_i)$ to the HH model described in (54)-(57) for 50000 time steps with $dt = 10^{-3}s$ and by collecting measurements and inputs $\mathcal{D} \triangleq \{u(t_0), y(t_0), \ldots, u(t_{N-1}), y(t_{N-1})\}$. We train our model on $\mathcal{D}$ with $P_{x_0} = 10^{-2} I_{d_x}, P_{\theta_0} = 10^{2} I_{d_\theta}$ $R_y = 10^{-10} I_{d_y}, Q_x = 10^{-5} I_{d_x}$ and $Q_\theta = 10^{-2} I_{d_\theta}$. At the beginning of each epoch, we solve the problem (66) of the Appendix D to get the initial condition. Final optimal parameters $\hat{\theta}(t_N)$ and initial condition $\hat{x}(t_0)$ are saved and collected at the end of training. Fig. 2 depicts the dynamics of the system $\hat{\theta}(t_N)$ generated according to the generative model described in Eq (4) starting from initial condition $\hat{x}(t_0)$. The lower right panel demonstrates the superiority of the proposed model at learning h_{gate}.

To demonstrate the robustness of the proposed approach to different dynamical regimes and showcase its capability of estimating accurate initial conditions, we perform an additional experiment. For this, we generate data $\mathcal{D}_T$ with $N = 50,000$ samples using the HH model with different initial conditions from the ones used during training. From this data, we reserve the first 100 samples for learning the initial condition before performing integration for the remaining $49,900$ samples. Then, using the learned model $\hat{\theta}(t_N)$ and the procedure described in Appendix D we obtained the initial condition $\hat{x}(t_{100})$ and obtained the proposed approach's solution. Figure 3 shows the evolution of the proposed approach attesting to its capability of both estimating accurate initial conditions and generalization to other dynamical regimes.

Cart-pole System

The cart-pole system is composed of a cart running on a track, with a freely swinging pendulum attached to it. The state of the system consists of the cart's position and velocity, and the pendulum's angle and angular velocity, while a control input u can be applied to the cart. We used the LQR (Prasad, Tyagi, and Gupta 2011) algorithm to learn a feedback controller that swings the pendulum and balances it in the inverted position in the middle of the track. The equations governing the ODE system are provided in (62)-(65) of the Appendix C.

We train our recursive model with the assumption that we don't know the equation corresponding to $\dot{\phi}$ governing dynamics of the cart-pole's angular rate. Therefore, we replace (63) and (65) with a two-layer LSTM

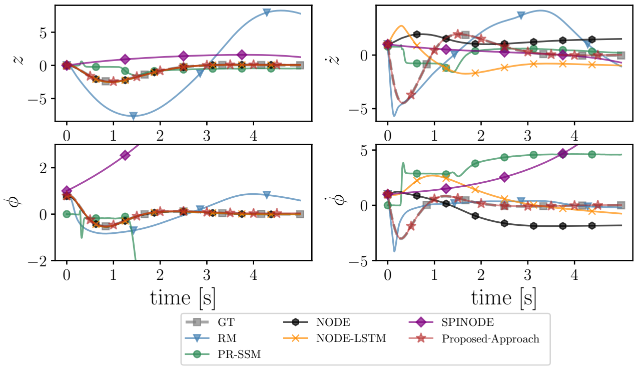

Figure 4: Learned state trajectories of the cart-pole system after training RM, PR-SSM, SPINODE, NODE, NODE-LSTM methods and the proposed approach. Results are compared to ground truth ODE system trajectory labeled as GT. We showed that the proposed approach can discern the true trajectory for the unmeasured states $\dot{z}$ and $\dot{\phi}$.

neural network when training with NODE-LSTM and a two-layer feedforward neural network for the rest of the benchmarks, with tanh activation function on each layer. We don't measure cart-pole's velocity $\dot{z}(t_i)$ and angular rate $\dot{\phi}(t_i)$, i.e., $y(t_i) = [z(t_i), \phi(t_i)]$. We generate our dataset by applying LQR balancing controller to the cart-pole described in Eqs (62)-(65) for 5000 time steps with $dt = 10^{-3}s$ and by collecting measurements and inputs $\mathcal{D} \triangleq \{u(t_0), y(t_0), \ldots, u(t_{N-1}), y(t_{N-1})\}$. We train our model on $\mathcal{D}$ with $P_{x_0} = 10^{-2}I_{d_x}, P_{\theta_0} = 10^2 I_{d_\theta}$ $R_y = 10^{-10}I_{d_y}, Q_x = 10^{-5}I_{d_x}$ and $Q_\theta = 10^{-2}I_{d_\theta}$. At the beginning of each epoch, we solve problem (66), Appendix D, to obtain initial conditions. The final optimal parameters $\hat{\theta}(t_N)$ and initial condition $\hat{x}(t_0)$ are saved and collected at the end of training. We qualitatively assess the performance of our model using the control sequence stored in $\mathcal{D}$ and optimal parameters $\hat{\theta}(t_N)$ according to the generative model described in (4) starting from initial condition $\hat{x}(t_0)$.

Table 1 shows that the proposed approach outperforms the competing algorithms with nRMSE value that is two to three orders of magnitude smaller when compared with competing methods. Analyzing the evolution of the latent states depicted in Figure 4, we notice that our proposed approach provides state trajectories that match the ground truth (GT) while the other methods fail to capture the true trajectory. PR-SSM presents acceptable trajectories of $\dot{z}$ and $\dot{z}$ but fails to learn ϕ and $\dot{\phi}$ trajectories. On the other hand, RM presents acceptable trajectories of ϕ and $\dot{\phi}$ but fails to learn z and $\dot{z}$ trajectories. Moreover, the NODE and NODE-LSTM successfully learn the measured ϕ and z trajectories but fail to learn correct trajectories of the unmeasured states $\dot{\phi}$ and $\dot{z}$. SPINODE, RM, and PR-SSM estimated state trajectories are much more inaccurate than the one provided by our proposed approach. The main reason for this inaccuracy is that trajectory generation is run using a pre-computing control sequence $\mathcal{U} \triangleq \{u(t_0), \ldots, u(t_{N-1}))\} \in \mathcal{D}$, hence any inaccuracy in the learned dynamics would cause the trajectories to go way off the ground truth (GT) due to the nonlinearity of the cart-pole system. This shows the challenging nature of the problem and the proposed approach's efficiency in learning challenging nonlinear dynamics. “In this context, the superior performance of the proposed approach is due to its alternating optimization approach, especially when compared with hybrid methods containing the same physics-informed terms, such as SPINODE and RM, since estimates of unmeasured states become available when optimizing θ.

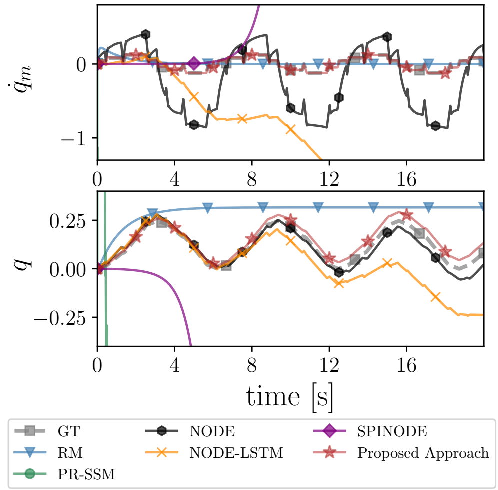

Figure 5: Learned state trajectories of EMPS after training RM, PR-SSM, SPINODE, NODE, NODE-LSTM methods and the proposed approach. Results are compared to ground truth ODE system trajectory labeled as GT. The proposed approach can discern the true trajectory for the unmeasured state $\dot{q}_m$.

Electro-mechanical positioning system

Here we evaluate the proposed approach on real data from an electro-mechanical positioning system described in (Janot, Gautier, and Brunot 2019). The training Dataset consists of system's of position, velocity, and control inputs used. The dataset consists of 24801 data points for each state and control input with $dt = 10^{-3}s$. Similarly to the HH and cart-pole systems, we train our proposed method using position and control inputs. we replace the velocity's dynamics with a feedforward neural network for all the benchmarks except NODE-LSTM where we used LSTM layers. We used two layers of 50 and 20 units respectively followed by a tanh activation function for all the benchmarks. Table 1 shows that the proposed approach outperforms the competing algorithms with nRMSE value one to three orders of magnitude smaller than the nRMSEs obtained by the competing methods. Analyzing the evolution of the latent states depicted in Figure 5, we notice that the proposed approach provides state trajectories that match the ground truth (GT) while SPINODE, PR-SSM, and RM collapse catastrophically. NODE learns the period of the hidden $\dot{q}_m$ signal but fails the capture its amplitude. The stiffness of $\dot{q}_m$ dynamics plays a role in these results since the sudden jumps shown in Figure 5 are hard to capture. This again demonstrates the robustness of the proposed approach, especially when compared to hybrid physics-based methods such as SPINODE and RM, demonstrating the better performance of the proposed alternating optimization procedure.

6 Conclusions

We proposed a novel recursive learning mechanism for NODE's to address the challenging task of learning the complex dynamics of ODE systems with partial measurements. Specifically, we constructed an alternating optimization procedure using Newton's method that sequentially finds optimal system latent states and model parameters. The resulting framework allows for efficient learning of missing ODEs when latent states are identifiable. Different from other competing methods, the proposed approach optimizes model parameters using latent states instead of measured data, leading to superior performance under the partial measurement setting. Experiments performed with five complex synthetic systems and one with real data provide evidence that our proposed method is capable of providing adequate solutions in very challenging scenarios.

References

Anil, R.; Gupta, V.; Koren, T.; Regan, K.; and Singer, Y. 2020. Scalable second order optimization for deep learning. *arXiv preprint arXiv:2002.09018*.

Bakarji, J.; Champion, K.; Kutz, J. N.; and Brunton, S. L. 2022. Discovering governing equations from partial measurements with deep delay autoencoders. *arXiv preprint arXiv:2201.05136*.

Bakarji, J.; Champion, K.; Nathan Kutz, J.; and Brunton, S. L. 2023. Discovering governing equations from partial measurements with deep delay autoencoders. *Proceedings of the Royal Society A*, 479(2276): 20230422.

Bell, B. M. 1994. The iterated Kalman smoother as a Gauss–Newton method. *SIAM Journal on Optimization*, 4(3): 626–636.

Bertsekas, D. P. 1996. Incremental least squares methods and the extended Kalman filter. *SIAM Journal on Optimization*, 6(3): 807–822.

Boltyanskiy, V.; Gamkrelidze, R. V.; Mishchenko, Y.; and Pontryagin, L. 1962. Mathematical theory of optimal processes.

Botev, A.; Ritter, H.; and Barber, D. 2017. Practical gauss-newton optimisation for deep learning. In *International Conference on Machine Learning*, 557–565. PMLR.

Buisson-Fenet, M.; Morgenthaler, V.; Trimpe, S.; and Di Meglio, F. 2022. Recognition Models to Learn Dynamics from Partial Observations with Neural ODEs. *Transactions on Machine Learning Research*.

Byrd, R. H.; Hansen, S. L.; Nocedal, J.; and Singer, Y. 2016. A stochastic quasi-Newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2): 1008–1031.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

De Brouwer, E.; Simm, J.; Arany, A.; and Moreau, Y. 2019. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. *Advances in neural information processing systems*, 32.

Demirkaya, A.; Imbiriba, T.; Lockwood, K.; Rampersad, S.; Alhajjar, E.; Guidoboni, G.; Danziger, Z.; and Erdogmus, D. 2021. Cubature Kalman Filter Based Training of Hybrid Differential Equation Recurrent Neural Network Physiological Dynamic Models. *43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*.

Doerr, A.; Daniel, C.; Schiegg, M.; Duy, N.-T.; Schaal, S.; Toussaint, M.; and Sebastian, T. 2018. Probabilistic recurrent state-space models. In *International conference on machine learning*, 1280–1289. PMLR.

Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Eleftheriadis, S.; Nicholson, T.; Deisenroth, M.; and Hensman, J. 2017. Identification of Gaussian process state space models. *Advances in neural information processing systems*, 30.

Ghanem, P.; Bicer, Y.; Erdogmus, D.; and Ramezani, A. 2021. Efficient Modeling of Morphing Wing Flight Using Neural Networks and Cubature Rules. *arXiv preprint arXiv:2110.01057*.

Goldfarb, D.; Ren, Y.; and Bahamou, A. 2020. Practical quasi-newton methods for training deep neural networks. *Advances in Neural Information Processing Systems*, 33: 2386–2396.

Gower, R.; Goldfarb, D.; and Richtárik, P. 2016. Stochastic block BFGS: Squeezing more curvature out of data. In *International Conference on Machine Learning*, 1869–1878. PMLR.

Guidoboni, G.; Harris, A.; Cassani, S.; Arciero, J.; Siesky, B.; Amireskandari, A.; Tobe, L.; Egan, P.; Januleviciene, I.; and Park, J. 2014. Intraocular pressure, blood pressure, and retinal blood flow autoregulation: a mathematical model to clarify their relationship and clinical relevance. *Investigative Ophthalmology & Visual Science*, 55(7): 4105–4118.

Gupta, V.; Koren, T.; and Singer, Y. 2018. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, 1842–1850. PMLR.

Hinton, G.; Srivastava, N.; and Swersky, K. 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8): 2.

Hodgkin, A. L.; and Huxley, A. F. 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117.

Humpherys, J.; Redd, P.; and West, J. 2012. A fresh look at the Kalman filter. *SIAM review*, 54(4): 801–823.

Ialongo, A. D.; Van Der Wilk, M.; Hensman, J.; and Rasmussen, C. E. 2019. Overcoming mean-field approximations in recurrent Gaussian process models. In *International Conference on Machine Learning*, 2931–2940. PMLR.

Imbiriba, T.; Demirkaya, A.; Duník, J.; Straka, O.; Erdogmus, D.; and Closas, P. 2022. Hybrid Neural Network Augmented Physics-based Models for Nonlinear Filtering. In *2022 25th International Conference on Information Fusion (FUSION)*, 1–6.

Janot, A.; Gautier, M.; and Brunot, M. 2019. Data set and reference models of EMPS. In *Nonlinear System Identification Benchmarks*.

Kaheman, K.; Kutz, J. N.; and Brunton, S. L. 2020. SINDy-PI: a robust algorithm for parallel implicit sparse identification of nonlinear dynamics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2242): 20200279.

Karniadakis, G. E.; Kevrekidis, I. G.; Lu, L.; Perdikaris, P.; Wang, S.; and Yang, L. 2021. Physics-informed machine learning. *Nature Reviews Physics*, 3(6): 422–440.

Kim, T. D.; Luo, T. Z.; Pillow, J. W.; and Brody, C. D. 2021. Inferring latent dynamics underlying neural population activity via neural differential equations. In *International Conference on Machine Learning*, 5551–5561. PMLR.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Mangan, N. M.; Brunton, S. L.; Proctor, J. L.; and Kutz, J. N. 2016. Inferring Biological Networks by Sparse Identification of Nonlinear Dynamics. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 2(1): 52–63.

Martens, J.; and Grosse, R. 2015. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, 2408–2417. PMLR.

Massaroli, S.; Poli, M.; Sonoda, S.; Suzuki, T.; Park, J.; Yamashita, A.; and Asama, H. 2021. Differentiable multiple shooting layers. *Advances in Neural Information Processing Systems*, 34: 16532–16544.

McHutchon, A. J.; et al. 2015. *Nonlinear modelling and control using Gaussian processes*. Ph.D. thesis, Citeseer.

Mokhtari, A.; and Ribeiro, A. 2014. RES: Regularized stochastic BFGS algorithm. *IEEE Transactions on Signal Processing*, 62(23): 6089–6104.

Norcliffe, A.; Bodnar, C.; Day, B.; Moss, J.; and Liò, P. 2021. Neural ode processes. *arXiv preprint arXiv:2103.12413*.

O'Leary, J.; Paulson, J. A.; and Mesbah, A. 2022. Stochastic physics-informed neural ordinary differential equations. *Journal of Computational Physics*, 468: 111466.

Peirson, A.; Amid, E.; Chen, Y.; Feinberg, V.; Warmuth, M. K.; and Anil, R. 2022. Fishy: Layerwise Fisher Approximation for Higher-order Neural Network Optimization. In *Has it Trained Yet? NeurIPS 2022 Workshop*.

Prasad, L. B.; Tyagi, B.; and Gupta, H. O. 2011. Optimal control of nonlinear inverted pendulum dynamical system with disturbance input using PID controller & LQR. In *2011 IEEE International Conference on Control System, Computing and Engineering*, 540–545. IEEE.

Psichogios, D. C.; and Ungar, L. H. 1992. A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10): 1499–1511.

Robbins, H.; and Monro, S. 1951. A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Rubanova, Y.; Chen, R. T.; and Duvenaud, D. K. 2019. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32.

Särkkä, S.; and Svensson, L. 2023. *Bayesian filtering and smoothing*, volume 17. Cambridge university press.

Schmidt, M. D.; Vallabhajosyula, R. R.; Jenkins, J.; Hood, J. E.; Soni, A. S.; Wikswo, J. P.; and Lipson, H. 2011. Automated refinement and inference of analytical models for metabolic networks. *Physical Biology*, 8: 055011.

Sholokhov, A.; Liu, Y.; Mansour, H.; and Nabi, S. 2023. Physics-informed neural ODE (PINODE): embedding physics into models using collocation points. *Scientific Reports*, 13(1): 10166.

Wan, E. A.; and Nelson, A. T. 2001. Dual extended Kalman filter methods. *Kalman filtering and neural networks*, 123–173.

Wang, Y.; Blei, D.; and Cunningham, J. P. 2021. Posterior collapse and latent variable non-identifiability. *Advances in Neural Information Processing Systems*, 34: 5443–5455.

Wieland, F.-G.; Hauber, A. L.; Rosenblatt, M.; Tönsing, C.; and Timmer, J. 2021. On structural and practical identifiability. *Current Opinion in Systems Biology*, 25: 60–69.

Xia, H.; Suliafu, V.; Ji, H.; Nguyen, T.; Bertozzi, A.; Osher, S.; and Wang, B. 2021. Heavy ball neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 34: 18646–18659.

Yildiz, C.; Heinonen, M.; and Lahdesmaki, H. 2019. ODE2VAE: Deep generative second order ODEs with Bayesian neural networks. *Advances in Neural Information Processing Systems*, 32.