Data-Adaptive Probabilistic Likelihood Approximation for Ordinary Differential Equations

Mohan Wu
University of Waterloo

Martin Lysy
University of Waterloo

Abstract

Estimating the parameters of ordinary differential equations (ODEs) is of fundamental importance in many scientific applications. While ODEs are typically approximated with deterministic algorithms, new research on probabilistic solvers indicates that they produce more reliable parameter estimates by better accounting for numerical errors. However, many ODE systems are highly sensitive to their parameter values. This produces deep local maxima in the likelihood function – a problem which existing probabilistic solvers have yet to resolve. Here we present a novel probabilistic ODE likelihood approximation, DALTON, which can dramatically reduce parameter sensitivity by learning from noisy ODE measurements in a data-adaptive manner. Our approximation scales linearly in both ODE variables and time discretization points, and is applicable to ODEs with both partially-unobserved components and non-Gaussian measurement models. Several examples demonstrate that DALTON produces more accurate parameter estimates via numerical optimization than existing probabilistic ODE solvers, and even in some cases than the exact ODE likelihood itself.

1 INTRODUCTION

Parameter estimation for ordinary differential equations (ODEs) is an important statistical and machine learning problem in the natural sciences and engineering. Typically this is accomplished via parameter searching algorithms which repeatedly solve the ODE at successive evaluations of the likelihood function. However, many ODE systems are hypersensitive to their input parameters, resulting in sharp local maxima in the likelihood function from which parameter search algorithms may fail to escape (Kohberger et al., 1978; Cao et al., 2011; Dass et al., 2017; Ma et al., 2021; Weber et al., 2018).

Since most ODEs do not have-closed form solutions, they must be approximated by numerical methods. While traditionally this has been done with deterministic algorithms (e.g., Butcher, 2008; Griffiths and Higham, 2010; Atkinson et al., 2009), a growing body of work in probabilistic numerics (Diaconis, 1988; Skilling, 1992; Hennig et al., 2015) indicates that probabilistic ODE solvers, which directly account for uncertainty in the numerical approximation, provide more reliable parameter estimates in ODE learning problems (Chkrebtii et al., 2016; Conrad et al., 2017). In particular, probabilistic solvers have the ability to condition on the observed data to guide the ODE solution, which can decrease sensitivity to model parameters (Chkrebtii et al., 2016; Yang et al., 2021; Schmidt et al., 2021; Tronarp et al., 2022). However, due to the increased complexity relative to deterministic solvers, the potential for probabilistic ODE solvers to reduce parameter hypersensitivity in a computationally efficient manner has yet to be fully realized.

Contributions Here we present a novel data-adaptive probabilistic ODE likelihood approximation (DALTON) which attempts to bridge this gap. Several examples will serve to illustrate the following key features of DALTON:

- *Scalability:* Building on the Bayesian filtering paradigm for probabilistic ODE solvers in Tronarp et al. (2019) and subsequent computational developments in Krämer et al. (2022), the DALTON algorithm scales linearly with both the number of time discretization points and the number of ODE variables.
- *Flexibility:* DALTON can be applied to arbitrary

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

order ODEs with partially-observed components – as can a number of related approximations (e.g., Schmidt et al., 2021; Yang et al., 2021; Tronarp et al., 2022). Unlike these, however, DALTON can also accommodate non-Gaussian measurement models, the utility of which is demonstrated in an application to infectious disease modeling (Section 5.2).

- *Robustness:* DALTON is able to overcome deep local modes in the ODE likelihood function that existing probabilistic and deterministic solvers cannot. This is demonstrated via insensitivity to initial values of a standard mode-finding algorithm for Bayesian estimation of the parameters of both oscillatory and chaotic ODE systems (examples 5.3 and 5.4), for which DALTON is shown to be more reliable than using the true ODE likelihood itself.

2 BACKGROUND

DALTON is designed to solve arbirary-order multivariable ODE systems which satisfy an initial value problem (IVP). For ease of presentation we focus on first-order systems. The extension to higher-order problems is described in Appendix A. For a multi-variable function $\boldsymbol{x}(t) = (x_1(t), \ldots, x_d(t))$, a first-order ODE-IVP is of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t), \qquad \boldsymbol{x}(0) = \boldsymbol{v}, \qquad t \in [0, T], \quad (1)$$

where $\dot{\boldsymbol{x}}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t)$ is the first derivative of $\boldsymbol{x}(t)$, $\boldsymbol{f}(\boldsymbol{x}(t), t) = (f_1(\boldsymbol{x}(t), t), \ldots, f_d(\boldsymbol{x}(t), t))$ is a nonlinear function, and $\boldsymbol{v} = \big(v_1, \ldots, v_d\big)$ is the initial value of $x(t)$ at time $t = 0$.

Unlike deterministic solvers, DALTON employs a probabilistic approach to solving (1) based on a well-established paradigm of Bayesian nonlinear filtering (Tronarp et al., 2019; Schober et al., 2019; Schmidt et al., 2021; Tronarp et al., 2022). This approach consists of putting a Gaussian Markov process on $\boldsymbol{x}(t)$ and its first $q - 1$ derivatives,

$$\boldsymbol{X}(t) = (\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t), \ldots, \tfrac{\mathrm{d}^{q-1}}{\mathrm{d}t^{q-1}}\boldsymbol{x}(t)), \quad (2)$$

and updating $\boldsymbol{X}(t)$ with information from the ODE-IVP (1) at time points $t = t_0, \ldots, t_N$, where $t_n = n \cdot \Delta t$ and $\Delta t = T/N$. Specifically, let $\boldsymbol{X}_n = \boldsymbol{X}(t_n)$ and consider the general indexing notation $\boldsymbol{X}_{m:n} = (\boldsymbol{X}_m, \ldots, \boldsymbol{X}_n)$. If $\boldsymbol{x}(t)$ is the solution to (1), we would have $\boldsymbol{Z}_n = \dot{\boldsymbol{x}}_n - \boldsymbol{f}(\boldsymbol{x}_n, t_n) = \boldsymbol{0}$. Based on this observation, Tronarp et al. (2019) consider a state-space model on $\boldsymbol{X}_n$ and $\boldsymbol{Z}_n$ of the form

$$\begin{aligned} \boldsymbol{X}_{n+1} \mid \boldsymbol{X}_n &\sim \mathrm{Normal}(\boldsymbol{Q}\boldsymbol{X}_n, \boldsymbol{R}) \\ \boldsymbol{Z}_n &\overset{\mathrm{ind}}{\sim} \mathrm{Normal}(\dot{\boldsymbol{x}}_n - \boldsymbol{f}(\boldsymbol{x}_n, t_n), \boldsymbol{V}), \end{aligned} \quad (3)$$

where $\boldsymbol{Q} = \boldsymbol{Q}(\Delta t)$ and $\boldsymbol{R} = \boldsymbol{R}(\Delta t)$ are determined by the Gaussian Markov process prior on $\boldsymbol{X}(t)$. The specific Gaussian Markov process prior used in this work is described in Appendix B. The stochastic ODE solution is then given by the posterior distribution $p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0})$ resulting from model (3).

As $N \to \infty$ and $\boldsymbol{V} \to \boldsymbol{0}$, the mode of the posterior distribution $p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0})$ gets arbitrarily close to the true ODE solution (Tronarp et al., 2021). However, this posterior distribution cannot be sampled from directly unless $\boldsymbol{f}(\boldsymbol{x}(t), t)$ is a linear function. Alternatives methods include Markov chain Monte Carlo (MCMC) sampling (Yang et al., 2021) and particle filtering (Tronarp et al., 2019). A less accurate but ostensibly much faster approach is to linearize (3), resulting in the working model

$$\begin{aligned} \boldsymbol{X}_{n+1} \mid \boldsymbol{X}_n &\sim \mathrm{Normal}(\boldsymbol{Q}\boldsymbol{X}_n, \boldsymbol{R}) \\ \boldsymbol{Z}_n &\overset{\mathrm{ind}}{\sim} \mathrm{Normal}(\dot{\boldsymbol{x}}_n + \boldsymbol{B}_n\boldsymbol{x}_n + \boldsymbol{a}_n, \boldsymbol{V}_n), \end{aligned} \quad (4)$$

where $\boldsymbol{V}_n$ is a tuning parameter used to capture the discrepancy between (4) and (3) with $\boldsymbol{V} = \boldsymbol{0}$. The benefit of the linearized model (4) is that it gives an approximation to (i) the posterior distribution $p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0})$, and (ii) the marginal likelihood $p(\boldsymbol{Z}_{0:N} = \boldsymbol{0})$ having linear complexity $\mathcal{O}(N)$ in the number of time discretization points, using standard Kalman filtering and smoothing techniques (Tronarp et al., 2019; Schober et al., 2019). Many linearization approaches can be found in Tronarp et al. (2019); Krämer et al. (2022) and different choices for $\boldsymbol{V}_n$ are discussed in Schober et al. (2019); Kersting et al. (2020b). Each determines the linearization parameters $(\boldsymbol{a}_n, \boldsymbol{B}_n, \boldsymbol{V}_n)$ at time $t = t_n$ in terms of the mean and variance of the Gaussian predictive distribution

$$p_{\mathrm{lin}}(\boldsymbol{x}_n \mid \boldsymbol{Z}_{0:n-1} = \boldsymbol{0}) \quad (5)$$

induced by the linearized working model (4) up to time $t = t_{n-1}$. Perhaps the simplest of these linearization schemes uses a zeroth-order Taylor approximation to $\boldsymbol{f}(\boldsymbol{x}_n, t_n)$ (Schober et al., 2019), with

$$\boldsymbol{a}_n = \boldsymbol{f}(\boldsymbol{\mu}_{n|n-1}, t_n), \quad \boldsymbol{B}_n = \boldsymbol{0}, \quad \boldsymbol{V}_n = \boldsymbol{0}, \quad (6)$$

where $\boldsymbol{\mu}_{n|n-1} = E_{\mathrm{lin}}[\boldsymbol{x}_n \mid \boldsymbol{Z}_{0:n-1} = \boldsymbol{0}]$. It has been shown that as $N \to \infty$, the mean and variance of the forward distribution $p_{\mathrm{lin}}(\boldsymbol{x}_n \mid \boldsymbol{Z}_{0:n} = \boldsymbol{0})$ induced by the linearization (6) converge, respectively, to the true ODE solution and zero (Kersting et al., 2020b).

3 METHODOLOGY

The parameter-dependent extension of the ODE-IVP (1) is of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}(t), t), \qquad \boldsymbol{x}(0) = \boldsymbol{v}_{\boldsymbol{\theta}}, \qquad t \in [0, T]. \quad (7)$$

The learning problem consists of estimating the unknown parameters of the model $\boldsymbol{\theta}$ which determine $\boldsymbol{x}(t)$ in (7) from noisy observations $\boldsymbol{Y}_{0:M} = (\boldsymbol{Y}_0, \dots, \boldsymbol{Y}_M)$, recorded at times $t = t'_0, \dots, t'_M$ under the measurement model

$$\boldsymbol{Y}_i \overset{\text{ind}}{\sim} p(\boldsymbol{Y}_i \mid \boldsymbol{x}(t'_i), \boldsymbol{\phi}), \tag{8}$$

with possibly unknown model parameters $\boldsymbol{\phi}$. In terms of the ODE solver discretization time points $t = t_0, \dots, t_N$, $N \geq M$, consider the mapping $n(\cdot)$ such that $t_{n(i)} = t'_i$, and the inverse mapping $i(n) = \max\{i : n(i) \leq n\}$. The Bayesian filtering model (3) is then augmented to account for noisy observations from (8) via (e.g., Schmidt et al., 2021)

$$\begin{aligned} \boldsymbol{X}_{n+1} \mid \boldsymbol{X}_n &\sim \text{Normal}(\boldsymbol{Q}_{\boldsymbol{\eta}} \boldsymbol{X}_n, \boldsymbol{R}_{\boldsymbol{\eta}}) \\ \boldsymbol{Z}_n &\overset{\text{ind}}{\sim} \text{Normal}(\dot{\boldsymbol{x}}_n - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_n, t_n), \boldsymbol{V}) \\ \boldsymbol{Y}_i &\overset{\text{ind}}{\sim} p(\boldsymbol{Y}_i \mid \boldsymbol{x}_{n(i)}, \boldsymbol{\phi}), \end{aligned} \tag{9}$$

where $\boldsymbol{\eta}$ are tuning parameters of the Gaussian Markov process prior. The likelihood function induced by the probabilistic solver corresponding to (9) for all parameters $\boldsymbol{\Theta} = (\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\eta})$ is given by (e.g., Kersting et al., 2020a; Tronarp et al., 2022)

$$\mathcal{L}(\boldsymbol{\Theta} \mid \boldsymbol{Y}_{0:M}) = p(\boldsymbol{Y}_{0:M} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}, \boldsymbol{\Theta}). \tag{10}$$

3.1 Gaussian Measurement Model

First suppose the observations of (8) consist of Gaussian noise,

$$\boldsymbol{Y}_i \overset{\text{ind}}{\sim} \text{Normal}(\boldsymbol{D}_i^{\boldsymbol{\phi}} \boldsymbol{x}_{n(i)}, \boldsymbol{\Omega}_i^{\boldsymbol{\phi}}), \tag{11}$$

where $\boldsymbol{D}_i^{\boldsymbol{\phi}}$ and $\boldsymbol{\Omega}_i^{\boldsymbol{\phi}}$ are (possibly $\boldsymbol{\phi}$-dependent) coefficient and variance matrices. To compute the likelihood (10), we begin with the identity

$$p(\boldsymbol{Y}_{0:M} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}) = \frac{p(\boldsymbol{Y}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0})}{p(\boldsymbol{Z}_{0:N} = \boldsymbol{0})}, \tag{12}$$

where we have omitted the dependence on $\boldsymbol{\Theta}$ for notational brevity. The denominator $p(\boldsymbol{Z}_{0:N} = \boldsymbol{0})$ on the right-hand side can be approximated using the Kalman recursions applied to the linearized state-space model (4). In fact, the same can be done for the numerator $p(\boldsymbol{Y}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0})$. That is, one begins by linearizing the measurement model (9) via

$$\begin{aligned} \boldsymbol{X}_{n+1} \mid \boldsymbol{X}_n &\sim \text{Normal}(\boldsymbol{Q}_{\boldsymbol{\eta}} \boldsymbol{X}_n, \boldsymbol{R}_{\boldsymbol{\eta}}) \\ \boldsymbol{Z}_n &\overset{\text{ind}}{\sim} \text{Normal}(\dot{\boldsymbol{x}}_n + \boldsymbol{B}_n \boldsymbol{x}_n + \boldsymbol{a}_n, \boldsymbol{V}_n) \\ \boldsymbol{Y}_i &\overset{\text{ind}}{\sim} \text{Normal}(\boldsymbol{D}_i^{\boldsymbol{\phi}} \boldsymbol{x}_{n(i)}, \boldsymbol{\Omega}_i^{\boldsymbol{\phi}}), \end{aligned} \tag{13}$$

and using any of the linearization approaches for the data-free setting in Section 2, but applied to the mean and variance of the Gaussian predictive distribution induced by (13),

$$p_{\text{lin}}(\boldsymbol{x}_n \mid \boldsymbol{Y}_{0:i(n-1)}, \boldsymbol{Z}_{0:n-1} = \boldsymbol{0}), \tag{14}$$

which conditions both on information from the ODE and the measurements up to time $t = t_{n-1}$. The complete DALTON algorithm for estimating $p(\boldsymbol{Y}_{0:M} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}, \boldsymbol{\Theta})$ is provided in Algorithm 1 of Appendix E, in terms of the standard Kalman filtering and smoothing recursions detailed in Appendix F.

3.2 Non-Gaussian Measurement Model

For the general measurement model (8), let $\boldsymbol{s}_{n(i)}$ and $\boldsymbol{u}_{n(i)}$ denote observed and unobserved components of $\boldsymbol{x}(t)$ at time $t = t'_i$, such that $\frac{\partial}{\partial \boldsymbol{u}_{n(i)}} p(\boldsymbol{Y}_i \mid \boldsymbol{x}_{n(i)}, \boldsymbol{\phi}) = \boldsymbol{0}$ and

$$p(\boldsymbol{Y}_i \mid \boldsymbol{x}_{n(i)}, \boldsymbol{\phi}) = \exp\{-g_i(\boldsymbol{Y}_i \mid \boldsymbol{s}_{n(i)}, \boldsymbol{\phi})\}. \tag{15}$$

In order to compute the likelihood (10), again omitting the dependence on $\boldsymbol{\Theta}$ we consider a different identity,

$$\begin{aligned} p(\boldsymbol{Y}_{0:M} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}) &= \frac{p(\boldsymbol{Y}_{0:M}, \boldsymbol{X}_{0:N} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0})}{p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Y}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0})} \\ &= \frac{p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}) \times \prod_{i=0}^{M} \exp\{-g_i(\boldsymbol{Y}_i \mid \boldsymbol{s}_{n(i)})\}}{p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Y}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0})}, \end{aligned} \tag{16}$$

where the identity holds for any value of $\boldsymbol{X}_{0:N}$. In the numerator of (16), $p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0})$ can be approximated using the Kalman smoothing algorithm applied to the data-free linearized model (4), whereas the product term is obtained via straightforward calculation of (15). As for the denominator of (16), we propose to approximate it by a multivariate normal distribution as follows. First, we note that

$$\begin{aligned} &\log p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Y}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0}) \\ &= \log p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}) - \sum_{i=0}^{M} h_i(\boldsymbol{s}_{n(i)}) + c_1, \end{aligned} \tag{17}$$

where $h_i(\boldsymbol{s}) = g_i^{\boldsymbol{\phi}}(\boldsymbol{Y}_i \mid \boldsymbol{s})$ and c_1 is constant with respect to $\boldsymbol{X}_{0:N}$. Next, we take a second-order Taylor expansion of $h_i(\boldsymbol{s}_{n(i)})$ about a value $\boldsymbol{s} = \hat{\boldsymbol{s}}_{n(i)}$ to be specified momentarily. After simplification this gives

$$h_i(\boldsymbol{s}) \approx -\frac{1}{2}(\boldsymbol{s} - \hat{\boldsymbol{Y}}_i)' \nabla^2 h_i(\hat{\boldsymbol{s}}_{n(i)})(\boldsymbol{s} - \hat{\boldsymbol{Y}}_i) + c_2, \tag{18}$$

where $\nabla h_i(\boldsymbol{s})$ and $\nabla^2 h_i(\boldsymbol{s})$ are the gradient and Hessian of $h_i(\boldsymbol{s})$, $\hat{\boldsymbol{Y}}_i = \hat{\boldsymbol{s}}_{n(i)} - \nabla^2 h_i(\hat{\boldsymbol{s}}_{n(i)})^{-1} \nabla h_i(\hat{\boldsymbol{s}}_{n(i)})$, and c_2 is constant with respect to $\boldsymbol{s}$. Substituting the quadratic approximation (18) into (17), we obtain an estimate of $p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Y}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0})$ equivalent to that of a

model with Gaussian observations,

$$\begin{aligned} \boldsymbol{X}_{n+1} \mid \boldsymbol{X}_n &\sim \text{Normal}(\boldsymbol{Q}_{\boldsymbol{\eta}}\boldsymbol{X}_n, \boldsymbol{R}_{\boldsymbol{\eta}}) \\ \boldsymbol{Z}_n &\overset{\text{ind}}{\sim} \text{Normal}(\dot{\boldsymbol{x}}_n - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_n, t_n), \boldsymbol{V}) \\ \hat{\boldsymbol{Y}}_i &\overset{\text{ind}}{\sim} \text{Normal}(\boldsymbol{s}_{n(i)}, [\nabla^2 h_i(\hat{\boldsymbol{s}}_{n(i)})]^{-1}). \end{aligned} \tag{19}$$

We may now linearize (19) exactly as in Gaussian setting of Section 3.1 to obtain an estimate

$$p_{\text{lin}}(\boldsymbol{X}_{0:N} \mid \hat{\boldsymbol{Y}}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0}), \tag{20}$$

which may be computed efficiently using the Kalman smoother. To complete the algorithm, there remains the choice $\boldsymbol{X}_{0:N}$ to plug into (16), and the choice of $\hat{\boldsymbol{s}}_{n(i)}$ about which to perform the Taylor expansion (18). For the latter, we use $\hat{\boldsymbol{s}}_{n(i)} = E_{\text{lin}}[\boldsymbol{s}_{n(i)} \mid \hat{\boldsymbol{Y}}_{0:i-1}, \boldsymbol{Z}_{0:n(i)-1} = \boldsymbol{0}]$, the predicted mean of the linearized model obtained from (19). For the former, we use $\boldsymbol{X}_{0:N} = E_{\text{lin}}[\boldsymbol{X}_{0:N} \mid \hat{\boldsymbol{Y}}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0}]$, the mean of (20), which can also be computed efficiently using the Kalman recursions. Full details of the DALTON algorithm for non-Gaussian measurements are provided in Algorithm 2 of Appendix E.

Other approximations to $p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Y}_{0:N}, \boldsymbol{Z}_{0:N} = \boldsymbol{0})$ include Laplace approximations (e.g., Amayo, 2006; Bui Quang et al., 2015) and posterior linearization methods (e.g., Särkkä and Svensson, 2023). These are likely more accurate than our second-order Taylor approximation (17)-(18) but incur a higher computational cost. Nevertheless, DALTON is conjectured to recover the exact ODE likelihood $\prod_{i=1}^{M} \exp\{-g_i(\boldsymbol{Y}_i \mid \boldsymbol{s}_{n(i)})\}$ as $N \to \infty$ and $\boldsymbol{V} \to \boldsymbol{0}$. Heuristically speaking, this is because (16) is precisely this quantity multiplied by the ratio of data-free and data-dependent posteriors at the true ODE solution. As $N \to \infty$ we expect the respective linearizations (4) and (20) to be the same, since the underlying state-space models (3) and (19) differ only in the finite amount of information provided by $\boldsymbol{Y}_{0:M}$, which becomes negligible relative to the infinite amount of information provided by $\boldsymbol{Z}_{0:N} = \boldsymbol{0}$.

Nevertheless, DALTON is conjectured to recover the true ODE likelihood as $N \to \infty$ and $\boldsymbol{V} \to \boldsymbol{0}$. Heuristically speaking, this is because our data-dependent linearization (20) is expected to reduce to the data-free linearization (4) as $N \to \infty$, since they differ only in the finite amount of information provided by $\boldsymbol{Y}_{0:M}$, which becomes negligible relative to the infinite amount of information provided by $\boldsymbol{Z}_{0:N} = \boldsymbol{0}$. Our linearized approximation to the identity (16) is thus expected to reduce to $\prod_{i=1}^{M} \exp\{-g_i(\boldsymbol{Y}_i \mid \hat{\boldsymbol{s}}_{n(i)})\}$, where the $\hat{\boldsymbol{s}}_{n(i)}$ are obtained from the linearized posterior mean $\hat{\boldsymbol{X}}_{0:N}$ of (20). Provided this converges to the true ODE solution – as in the data-free setting – the true ODE likelihood is recovered. A rigorous derivation of this informal argument is the subject of ongoing work.

4 RELATED WORK

The Bayesian filtering paradigm (3) was formulated by Tronarp et al. (2019). Speed and stability improvements on the linearizations presented therein were developed in Krämer and Hennig (2020); Krämer et al. (2022). Various convergence properties may be found in Chkrebtii et al. (2016); Schober et al. (2019); Kersting et al. (2020b).

Gradient matching with Gaussian processes (Calderhead et al., 2009) is an early precursor to the Bayesian filtering paradigm (3). Similar in spirit to deterministic collocation solvers (e.g., Ramsay et al., 2007), a conceptual limitation of this approach is the lack of a generative model on both the Gaussian process $\boldsymbol{X}(t)$ and the observed data $\boldsymbol{Y}_{0:M}$ (Barber and Wang, 2014), leading to parameter identifiability issues (Macdonald et al., 2015; Wenk et al., 2019). Moreover, many gradient-matching approaches involve computationally-intensive methods such as MCMC (Barber and Wang, 2014; Lazarus et al., 2018) and approximate Bayesian computation (ABC) (Ghosh et al., 2017), though computationally efficient variational approximations have been developed as well (e.g., Dondelinger et al., 2013; Gorbach et al., 2017; Wenk et al., 2019).

Active uncertainty calibration (Kersting and Hennig, 2016) formulates the ODE solver as a Gaussian process filtering problem. Not only does this approach allow to naturally extend many deterministic ODE solvers (e.g., Schober et al., 2014; Teymur et al., 2016; Schober et al., 2019; Bosch et al., 2022), it also admits analytic calculations which scale linearly in the number of discretization points N when a Markov Gaussian process prior is employed. Active uncertainty calibration is operationally equivalent to linearization of the Bayesian filtering model (3), though the latter has the conceptual benefit of separating the desired solution posterior $p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Z}_{0:N})$ from the linearization (4) used to approximate it.

Within the Bayesian filtering framework, we highlight the following approaches in which the ODE solver directly conditions on the observed data:

Chkrebtii et al. (2016) uses active uncertainty calibration to stochastically estimate the optimal linearization parameter $\boldsymbol{a}_n$ in the sense of Kersting and Hennig (2016); Tronarp et al. (2019). However, the nondeterministic $\boldsymbol{a}_n$ cannot be analytically integrated out, such that inference for $\boldsymbol{\Theta}$ is conducted by sampling from the joint posterior $p(\boldsymbol{\Theta}, \boldsymbol{X}_{0:N} \mid \boldsymbol{Y}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0})$ uses MCMC techniques.

MAGI (Yang et al., 2021) uses a closed-form expression for the posterior distribution $p(\boldsymbol{\Theta}, \boldsymbol{X}_{0:N} \mid \boldsymbol{Y}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0})$ resulting from the Bayesian filter-

ing model (9) with "perfect" ODE information $\boldsymbol{V} = \boldsymbol{0}$. Again this requires sampling from the joint posterior using MCMC techniques.

Schmidt et al. (2021) uses a data-driven extended Kalman filter similar to ours, but assumes time-varying parameters for which they give a Gaussian process prior, thus treating them just like the latent ODE process in the nonlinear state-space model. However, their approach cannot be used for constant parameter inference, since linearization in that context is generally leads to biased or even divergent estimates (Ljung, 1979).

Fenrir (Tronarp et al., 2022) extends an approach developed in Kersting et al. (2020a). It begins by using the data-free linearization of (4) to estimate $p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}, \boldsymbol{\theta}, \boldsymbol{\eta})$. This posterior can be sampled from using a (non-homogeneous) Markov chain going backwards in time,

$$\begin{aligned} \boldsymbol{X}_N &\sim \text{Normal}(\boldsymbol{b}_N, \boldsymbol{C}_N) \\ \boldsymbol{X}_n \mid \boldsymbol{X}_{n+1} &\sim \text{Normal}(\boldsymbol{A}_n \boldsymbol{X}_n + \boldsymbol{b}_n, \boldsymbol{C}_n), \end{aligned} \tag{21}$$

where the coefficients $\boldsymbol{A}_{0:N-1}$, $\boldsymbol{b}_{0:N}$, and $\boldsymbol{C}_{0:N}$ are derived using the Kalman recursions (Tronarp et al., 2022). Next, Fenrir assumes that Gaussian observations are added to the model, for which the marginal likelihood

$$\begin{aligned} &p(\boldsymbol{Y}_{0:M} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}, \boldsymbol{\Theta}) \\ &= \int p(\boldsymbol{Y}_{0:N} \mid \boldsymbol{X}_{0:N}, \boldsymbol{\Theta}) p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}, \boldsymbol{\Theta}) \, \mathrm{d}\boldsymbol{X}_{0:N} \end{aligned} \tag{22}$$

is computed using a Kalman filter on the backward pass of (21). Fenrir is fast, accurate, and compares favorably to both the fast gradient-based solver of Wenk et al. (2019) and various deterministic ODE solvers (Tronarp et al., 2022). The key difference between DALTON and Fenrir is that the latter linearizes *before* adding the observations to the model (linearize-then-observe), whereas the former does so *after* (observe-then-linearize). Moreover, DALTON can be applied to non-Gaussian errors whereas Fenrir cannot.

Computational Scaling DALTON scales linearly in both N, the number of time discretization points, and d, the dimension of the ODE solution $\boldsymbol{x}(t)$. The former is due to the Markov structure of the prior on $\boldsymbol{X}(t)$, whereas the latter is achieved by using the blockwise linearization of Krämer et al. (2022) (details in Appendix C).

For the Gaussian measurement model (11), the operation count of DALTON and Fenrir is roughly the same, with Fenrir doing one forward and one backward pass through the N discretization points, and DALTON doing one forward pass for each of $p(\boldsymbol{Z}_{0:N} = \boldsymbol{0} \mid \boldsymbol{\Theta})$ and $p(\boldsymbol{Y}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0} \mid \boldsymbol{\Theta})$ in (12). For non-Gaussian measurements, DALTON requires roughly twice as many operations as in the Gaussian case, since the former requires one forward and backward pass for each of $p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}, \boldsymbol{\theta}, \boldsymbol{\eta})$ and $p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Y}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0}, \boldsymbol{\Theta})$ in (16).

5 EXAMPLES

We now evaluate the performance of DALTON in several numerical examples of parameter learning. We proceed with a Bayesian approach by postulating a prior distribution $\pi(\boldsymbol{\Theta})$ on the full set of parameters $\boldsymbol{\Theta} = (\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\eta})$, which combined with (10) gives the DALTON posterior

$$\begin{aligned} p(\boldsymbol{\Theta} \mid \boldsymbol{Y}_{0:M}) &\propto \pi(\boldsymbol{\Theta}) \times \mathcal{L}(\boldsymbol{\Theta} \mid \boldsymbol{Y}_{0:M}) \\ &= \pi(\boldsymbol{\Theta}) \times p(\boldsymbol{Y}_{0:M} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}, \boldsymbol{\Theta}). \end{aligned} \tag{23}$$

In all our examples we take the measurement model parameter $\boldsymbol{\phi}$ to be known.

While (23) can be readily sampled from using MCMC techniques, Bayesian inference can also be achieved by way of a Laplace approximation (e.g., Gelman et al., 2013). Namely, $p(\boldsymbol{\Theta} \mid \boldsymbol{Y}_{0:M})$ is approximated by a multivariate normal distribution,

$$\boldsymbol{\Theta} \mid \boldsymbol{Y}_{0:M} \approx \text{Normal}(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{V}}_{\boldsymbol{\Theta}}), \tag{24}$$

where $\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} \log p(\boldsymbol{\Theta} \mid \boldsymbol{Y}_{0:M})$ and $\hat{\boldsymbol{V}}_{\boldsymbol{\Theta}} = -\big[\frac{\partial^2}{\partial \boldsymbol{\Theta} \partial \boldsymbol{\Theta}'} \log p(\hat{\boldsymbol{\Theta}} \mid \boldsymbol{Y}_{0:M})\big]^{-1}$. The Laplace approximation is a popular tool for Bayesian machine learning applications (MacKay, 1992; Gianniotis, 2019), typically requiring at least an order of magnitude fewer evaluations of $p(\boldsymbol{\Theta} \mid \boldsymbol{Y}_{0:M})$ than full Bayesian inference via MCMC. Our Python implementation of DALTON[1] uses the **JAX** library (Bradbury et al., 2018) for automatic differentiation and just-in-time (JIT) compilation to obtain a very fast implementation of the Laplace approximation (24).

Data for the examples are simulated using a high-accuracy deterministic solver for the ODE-IVP (7), namely, the Dormand-Prince 8/7 solver consisting of an 8th order Runge-Kutta method with 7th order step size adaptation (hereafter RKDP) (Prince and Dormand, 1981). In the following sections, we compare parameter inference using the DALTON Laplace approximation to three competing probabilistic ODE likelihood methods: (i) a Laplace approximation with the Fenrir likelihood, and full Bayesian inference via MCMC for the (ii) MAGI and (iii) Chkrebtii et al. (2016) methods (hereafter Chkbretii). We also compare to (iv) a Laplace approximation using the deterministic RKDP solver,

[1] `https://github.com/mlysy/rodeo`

which we assume to provide the true ODE solution. We use our own implementations of DALTON, Fenrir, and Chkbretii, the MAGI implementation provided by Wong et al. (2023)[2], and the RKDP implementation provided by Kidger (2021)[3]. The Chkbretii and MAGI algorithms are run for 100,000 and 10,000 iterations each (post burn-in), which produced effective samples sizes ranging between 100-5000. Further implementation details for the parameter learning methods are provided in Appendix D. Additionally, we provide comparisons to full MCMC with both the DALTON likelihood and the true RKDP likelihood in Appendix G.

5.1 FitzHugh-Nagumo Model

The FitzHugh-Nagumo (FN) model (FitzHugh, 1961; Nagumo et al., 1962) is a two-state ODE on $\boldsymbol{x}(t) = (V(t), R(t))$, in which $V(t)$ describes the evolution of the neuronal membrane voltage and $R(t)$ describes the activation and deactivation of neuronal channels. The FN ODE is given by

$$\begin{aligned} \dot{V}(t) &= c\Big(V(t) - \frac{V(t)^3}{3} + R(t)\Big), \\ \dot{R}(t) &= -\frac{V(t) - a + bR(t)}{c}. \end{aligned} \tag{25}$$

The model parameters are $\boldsymbol{\theta} = (a, b, c, V(0), R(0))$ with $a, b, c > 0$. These are to be learned from the measurement model

$$\boldsymbol{Y}_i \overset{\text{ind}}{\sim} \text{Normal}(\boldsymbol{x}(t_i), \phi^2 \cdot \boldsymbol{I}_{2\times 2}), \tag{26}$$

where $t_i = i$ with $i = 0, 1, \ldots 40$ and $\phi^2 = 0.005$. The ODE and noisy observations are displayed in Figure 1.

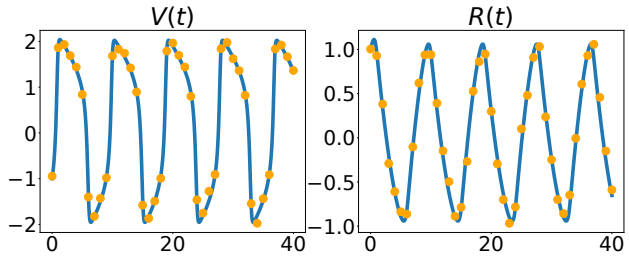

Figure 1: ODE and noisy observations for the FN model.

Figure 2 displays a subset of the DALTON, Fenrir, Chkbretii, and MAGI posteriors for data simulated with true parameter values $\boldsymbol{\theta} = (0.2, 0.2, 3, -1, 1)$, at different values of the step size $\Delta t = T/N$. In Figure 2, all posterior distributions are virtually identical at smaller step sizes. At larger step sizes, MAGI

[2] https://github.com/wongswk/magi
[3] https://github.com/patrick-kidger/diffrax

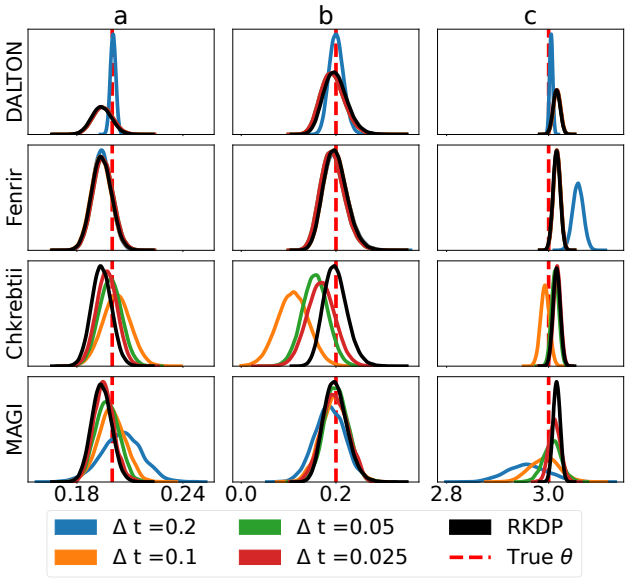

Figure 2: Parameter posteriors for the FN model using various ODE solvers.

and Chkbretii are further from the RKDP posterior than either DALTON or Fenrir, which Chkbretii failing to converge at the largest step size $\Delta t = 0.2$. For $\Delta t = 0.2$, DALTON and Fenrir posteriors are similar, with the notable exception of the posterior for c, where DALTON covers the true parameter value but Fenrir does not. This is because c controls the period of the quasi-oscillatory behavior of the FN model exhibited in Figure 1 (e.g., Rocsoreanu et al., 2012), which is a prime example of when incorporating information from the observed data on the forward pass (DALTON) rather than the backward pass (Fenrir) is most useful.

Table 1 displays the timing for each of the probabilistic posterior estimates in Figure 2, with the number of function-and-gradient evaluations required for the Fenrir, DALTON and RKDP Laplace approximations displayed in parentheses. The DALTON and Fenrir Laplace approximations are 1-2 orders of magnitude faster than either of the methods requiring MCMC and have comparable speeds to RKDP-Laplace. Although DALTON employs twice as many vector field evaluations as Fenrir, this does not seem to be the computational bottleneck as shown in Table 1.

The full set of posteriors is presented in Figure 9 of Appendix G, along with exact posteriors for DALTON and RKDP obtained via MCMC. These are extremely similar to their Laplace counterparts but 2-3 orders of magnitude slower to compute. Not only does this show that the Laplace approximation is a viable competitor to MCMC in this setting, it also shows that the DALTON likelihood approximation is accurate over

Table 1: Time spent on the parameter inference results in seconds.

Method	Step Size (Δt)			
	0.2	0.1	0.05	0.025
DALTON	0.5 (628)	0.07 (27)	0.09 (16)	0.2 (15)
Fenrir	0.6 (585)	0.1 (34)	0.1 (18)	0.2 (17)
Chkbretii	-	92	189	372
MAGI	222	418	742	1349
RKDP		.04 (79)		

the whole parameter range – not just near the posterior mode.

5.2 SEIRAH model

The SEIRAH model is a six-compartment epidemiological model used describe Covid-19 dynamics by Prague et al. (2020). The six compartments of a given population are: susceptible (S_t), latent (E_t), ascertained infectious (I_t), removed (R_t), unascertained infectious (A_t), and hospitalized (H_t). The SEIRAH model is

$$\begin{aligned} \dot{S}_t &= -\frac{bS_t(I_t+\alpha A_t)}{N}, & \dot{E}_t &= \frac{bS_t(I_t+\alpha A_t)}{N} - \frac{E_t}{D_e}, \\ \dot{I}_t &= \frac{rE_t}{D_e} - \frac{I_t}{D_q} - \frac{I_t}{D_I}, & \dot{R}_t &= \frac{I_t+A_t}{D_I} + \frac{H_t}{D_h}, \\ \dot{A}_t &= \frac{(1-r)E_t}{D_e} - \frac{A_t}{D_I}, & \dot{H}_t &= \frac{I_t}{D_q} - \frac{H_t}{D_h}, \end{aligned} \tag{27}$$

where $N = S_t + E_t + I_t + R_t + A_t + H_t$ is fixed and so is $D_h = 30$ (Prague et al., 2020). In this case ODE variables are both partially unobserved and subject to non-Gaussian measurement errors. That is, for

$$\tilde{I}(t) = \frac{rE_t}{D_e}, \quad \tilde{H}(t) = \frac{I_t}{D_q}, \tag{28}$$

the measurement model is

$$Y_i^{(K)} \overset{\text{ind}}{\sim} \text{Poisson}(\tilde{K}(t_i)), \qquad K \in \{I, H\}, \tag{29}$$

where $t_i = i$ days with $i = 0, \ldots, 60$. The parameters to be estimated are $\boldsymbol{\theta} = (b, r, \alpha, D_e, D_q, E_0, I_0)$, which we set to $\boldsymbol{\theta} = (2.23, 0.034, 0.55, 5.1, 2.3, 1.13, 15,492, 21,752)$. The remaining ODE initial values are set to $S_0 = 63,884,630$, $R_0 = 0$, $A_0 = 618,013$, and $H_0 = 13,388$, as reported in Prague et al. (2020).

Since the measurement model is non-Gaussian, Fenrir cannot be used. Moreover, the MAGI implementation provided by Wong et al. (2023) is restricted to Gaussian measurement models. Therefore, Figure 3 displays posterior estimates for a subset of SEIRAH parameters only for DALTON, Chkbretii, and RKDP. At all but the lowest step size, the Laplace approximations for DALTON and the true likelihood (RKDP) are indistinguishable. In contrast, the Chkbretii approximation has not converged to that of RKDP even at the smallest step size $\Delta t = 0.05$.

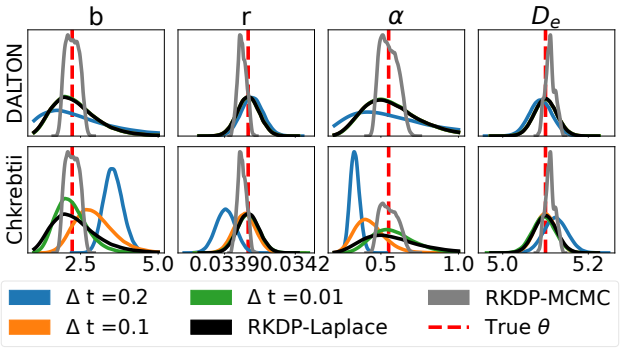

Figure 3: Parameter posteriors for the SEIRAH model for the DALTON and Chkbretii methods.

The full set of posteriors is presented in Figure 10 of Appendix G, along with true posteriors for DALTON and RKDP. In this case, the true RKDP posterior is much narrower than its Laplace approximation. However, it is almost indistinguishable from the true posterior of DALTON, once again highlighting the accuracy of the approximation away from the posterior mode.

5.3 Fast Oscillation Model

The following second-order ODE is known for having many local maxima in its parameter likelihood function (Tronarp et al., 2022):

$$\ddot{x}(t) = -\frac{g}{L}\sin(x(t)), \tag{30}$$

where $\ddot{x}(t) = \frac{\mathrm{d}^2}{\mathrm{d}t^2}x(t)$ and $g = 9.81$ is the gravity constant. The model parameters are $\boldsymbol{\theta} = (L, x(0), \dot{x}(0))$ with $L > 0$, which are to be learned from the measurement model

$$Y_i \overset{\text{ind}}{\sim} \text{Normal}(\dot{x}(t_i), \phi^2), \tag{31}$$

where $t_i = i$, $i = 0, 1, \ldots 10$, and $\phi^2 = 0.1$. (The extension of the Bayesian filtering model (3) to higher-order systems is presented in Appendix A.) Following Tronarp et al. (2022), the true parameter value is set to $\boldsymbol{\theta} = (1, 0, \pi/2)$, and the initial value for each parameter learning algorithm (i.e., mode-finding for Laplace; otherwise MCMC) is set to $\boldsymbol{\theta}_0 = (5, 0, \pi/2)$ – a poorly-chosen initial guess for L.

Figure 4 displays posterior estimates for the four probabilistic likelihood approximations and for the true likelihood given by RKDP. Chkbretii, MAGI, and RKDP

give very poor estimates of L. In contrast, DALTON and Fenrir provide reasonable estimates of all parameters. This experiment confirms the finding of Tronarp et al. (2022) that mode-finding via Fenrir dominates other probabilistic likelihood approximations – and even mode-finding with true likelihood itself – when the latter is highly multimodal. The same is now shown for mode-finding via DALTON.

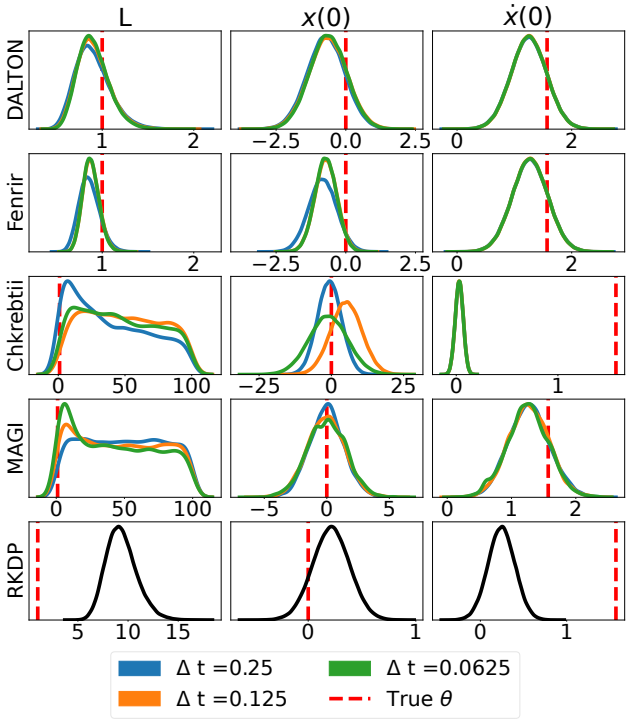

Figure 4: Parameter posteriors for the second-order ODE using various ODE solvers where the initial parameter value is $\boldsymbol{\theta}_0 = (5, 0, \pi/2)$.

5.4 Lorenz63 Model

The Lorenz63 ODE (Lorenz, 1963) models the dynamics of a fluid layer induced by a warm temperature above and a cool temperature below. It is a classical example of a chaotic system, composed of three variables $\boldsymbol{x}(t) = (x(t), y(t), z(t))$ satisfying

$$\begin{aligned} \dot{x}(t) &= \alpha(y(t) - x(t)), \\ \dot{y}(t) &= x(t)(\rho - z(t)) - y(t), \\ \dot{z}(t) &= x(t)y(t) - \beta z(t). \end{aligned} \tag{32}$$

The Lorenz63 model contains six parameters $\boldsymbol{\theta} = (\rho, \alpha, \beta, x(0), y(0), z(0))$ with $\rho, \alpha, \beta > 0$. The measurement error model is

$$\boldsymbol{Y}_i \overset{\text{ind}}{\sim} \text{Normal}(\boldsymbol{x}(t_i), \phi^2 \cdot \boldsymbol{I}_{3\times 3}), \tag{33}$$

where $t_i = i$, $i = 0, 1, \ldots, 20$, and $\phi^2 = 0.005$. The true parameter values are set to $\boldsymbol{\theta} = (28, 10, 8/3, -12, -5, 38)$.

Figure 5 displays various estimates of the system variable $x(t)$ assuming the model parameters are known (the remaining components are shown in Figure 11 of Appendix G). Since Lorenz63 is even more sensitive to parameter values than the fast-oscillations model (30), Fenrir and DALTON are the only data-driven estimators considered. Also plotted is the true solution given by RKDP, and the data-free probabilistic solution using the linearization of (3) common to both Fenrir and DALTON (described in Appendix C). Like the data-free solver, Fenrir struggles with the chaotic nature of the Lorenz63 model even at very high resolution ($\Delta t = 10^{-5}$), being only able to incorporate information from the data on the backward pass. In contrast, DALTON effectively uses the observations on the forward pass to produce a solution close to the ground truth at much lower resolution.

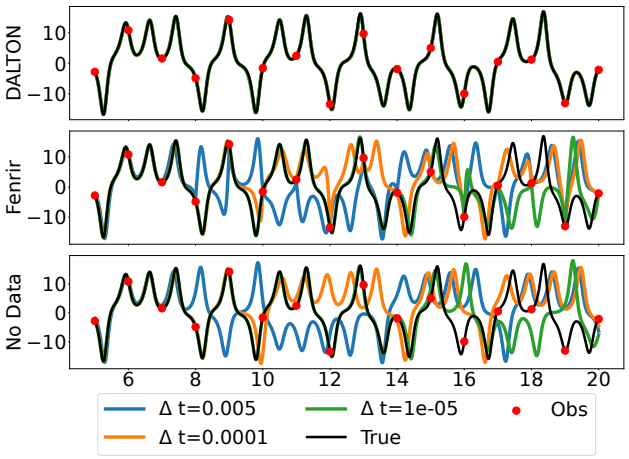

Figure 5: $x(t)$ for the Lorenz63 model calculated using various solvers.

Figure 6 displays a subset of the DALTON, Fenrir, and RKDP posteriors (remaining plots in Figure 12 of Appendix G), with mode-finding routines initialized at two different values of $\boldsymbol{\theta}$: at and slightly away from the true parameter values. When initialized at the true $\boldsymbol{\theta}$, the Fenrir and RKDP posteriors cover the true parameter values with vanishingly little uncertainty. However, when their mode-finding algorithms are initialized slightly away from the true $\boldsymbol{\theta}$, uncertainty remains small but the coverage drops to zero. In contrast, the DALTON posteriors always cover the true parameter values and don't depend on the mode-finding algorithm's initial value. This appears to be due to deep local optima in the Fenrir and RKDP likelihoods which DALTON successfully removes.

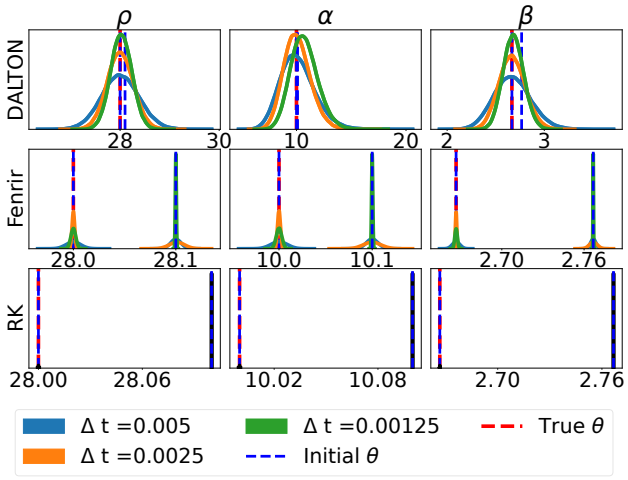

Figure 6: Parameter posteriors for the Lorenz63 model using various ODE solvers at two initial parameter values.

To investigate this claim, we modify the parameter learning problem so that $t_i = i/10$ with $i = 0, 1, \ldots, 200$ and with initial values $(x(0), y(0), z(0))$ assumed to be known. Figure 7 displays the posterior modes of α, ρ, and β against different initializations of the mode-finding algorithms for DALTON, Fenrir and RKDP. When initialized far from the ground truth, Fenrir and RKDP fail to converge to the correct parameter values. Since their posterior uncertainty in all cases is negligible, even the true Laplace posterior produced by RKDP is unusable for parameter inference. In contrast, DALTON is able to converge to the true parameters at a wide range of initial values, while retaining the parameter coverage exhibited in Figure 6.

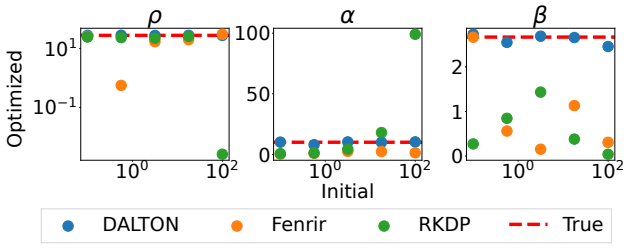

Figure 7: Initial and optimized parameter values for the Lorenz63 model using various ODE solvers.

Figure 8 displays the DALTON solution posterior at $\Delta t = 0.00125$ for component $x(t)$ (the remaining components are displayed in Figure 13 of Appendix G). Care must be taken in computing this posterior, since exact ODEs simulated from the DALTON parameter posteriors in Figure 6 look nothing like the true solution, due to extreme parameter sensitivity. Instead, the solution posterior is taken with respect to the probabilistic model (9), i.e., each curve in Figure 8 is obtained by first drawing $\boldsymbol{\Theta}$ from the DALTON posterior in Figure 6, then drawing $x_{0:N}$ from the linearized approximation (20) to $p(\boldsymbol{X}_{0:N} \mid \boldsymbol{\Theta}, \boldsymbol{Y}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0})$. Figure 8 indicates that these solution posteriors closely resemble the true ODE.

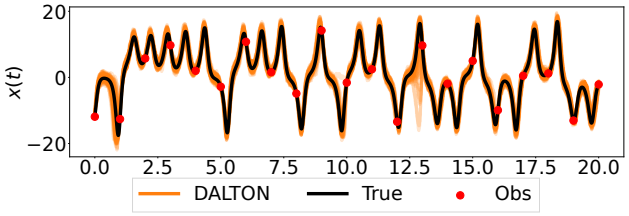

Figure 8: ODE posteriors of $x(t)$ for the Lorenz63 model using DALTON with $\Delta t = 0.00125$.

6 CONCLUSION

We present DALTON, a probabilistic approximation to the intractable likelihood of ODE-IVP problems. DALTON scales linearly in both time discretization points and ODE variables, accommodating both partially-observed components and non-Gaussian measurement models. By incorporating information from the observed data in an online maner, DALTON can greatly reduce parameter sensitivity in many ODE systems. It is thus an excellent candidate for parameter learning via mode-finding algorithms, outperforming existing probabilistic ODE likelihood approximations, and in cases of extreme parameter sensitivity, even the exact ODE likelihood itself.

Despite empirical evidence of desirable performance, DALTON presently lacks theoretical convergence guarantees. It should be noted, however, that the MAGI and Fenrir methods do not have complete theoretical guarantees either: only maximum *a posteriori* (MAP) convergence is known of the former (Tronarp et al., 2021), and forward-pass convergence of the latter (Kersting et al., 2020b). Another limitation of DALTON is determining the appropriate step size. A possible direction is to do this adaptively as described in Schober et al. (2019), though it is unclear how the DALTON ratio terms in (12) and (16) will behave with a variable number of terms, nor how to efficiently backpropagate derivatives through the ODE parameters in that case. There is also potential to explore the effectiveness of DALTON in estimating the parameters of stiff ODE systems, and to extend it to more complex boundary conditions.

References

Amayo, E. O. (2006). Construction of nonlinear filter algorithms using the saddlepoint approximation. Master's thesis, Massachusetts Institute of Technology.

Atkinson, K., Han, W., and Stewart, D. E. (2009). *Numerical Solution of Ordinary Differential Equations*. John Wiley & Sons.

Barber, D. and Wang, Y. (2014). Gaussian processes for Bayesian estimation in ordinary differential equations. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 1485–1493. PMLR.

Blondel, M., Berthet, Q., Cuturi, M., Frostig, R., Hoyer, S., Llinares-Lopez, F., Pedregosa, F., and Vert, J.-P. (2022). Efficient and modular implicit differentiation. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 5230–5242. Curran Associates, Inc.

Bosch, N., Tronarp, F., and Hennig, P. (2022). Pick-and-mix information operators for probabilistic ODE solvers. In Camps-Valls, G., Ruiz, F. J. R., and Valera, I., editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151, pages 10015–10027. PMLR.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

Bui Quang, P., Musso, C., and Le Gland, F. (2015). The Kalman Laplace filter: a new deterministic algorithm for nonlinear bayesian filtering. In *2015 18th International Conference on Information Fusion*, pages 1566–1573.

Butcher, J. C. (2008). *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons.

Calderhead, B., Girolami, M., and Lawrence, N. D. (2009). Accelerating Bayesian inference over nonlinear differential equations with Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 217–224. Curran Associates, Inc.

Cao, J., Wang, L., and Xu, J. (2011). Robust estimation for ordinary differential equation models. *Biometrics*, 67(4):1305–1313.

Chkrebtii, O. A., Campbell, D. A., Calderhead, B., and Girolami, M. A. (2016). Bayesian solution uncertainty quantification for differential equations. *Bayesian Analysis*, 11(4):1239–1267.

Conrad, P. R., Girolami, M., Särkkä, S., Stuart, A., and Zygalakis, K. (2017). Statistical analysis of differential equations: introducing probability measures on numerical solutions. *Statistics and Computing*, 27(4):1065–1082.

Dass, S., Lee, J., Lee, K., and Park, J. (2017). Laplace based approximate posterior inference for differential equation models. *Statistics and Computing*, 27(3):679–698.

Diaconis, P. (1988). Bayesian numerical analysis. In Berger, J. and Gupta, S., editors, *Statistical Decision Theory and Related Topics IV*, volume 1, pages 163–175. Springer-Verlag, New York.

Dondelinger, F., Husmeier, D., Rogers, S., and Filippone, M. (2013). ODE parameter inference using adaptive gradient matching with Gaussian processes. In Carvalho, C. M. and Ravikumar, P., editors, *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31, pages 216–228. PMLR.

FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6):445–466.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian Data Analysis*. Chapman & Hall, New York, NY, 3rd edition.

Ghosh, S., Dasmahapatra, S., and Maharatna, K. (2017). Fast approximate Bayesian computation for estimating parameters in differential equations. *Statistics and Computing*, 27(1):19–38.

Gianniotis, N. (2019). Mixed variational inference. In *2019 International Joint Conference on Neural Networks*, pages 1–8. IEEE.

Gorbach, N. S., Bauer, S., and Buhmann, J. M. (2017). Scalable variational inference for dynamical systems. In *Advances in Neural Information Processing Systems*, pages 4809–4818. Curran Associates, Inc.

Griffiths, D. F. and Higham, D. J. (2010). *Numerical Methods for Ordinary Differential Equations: Initial Value Problems*. Springer Science & Business Media.

Hennig, P., Osborne, M. A., and Girolami, M. (2015). Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2179):20150142.

Kersting, H. and Hennig, P. (2016). Active uncertainty calibration in Bayesian ODE solvers. In *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence*, pages 309–318.

Kersting, H., Krämer, N., Schiegg, M., Daniel, C., Tiemann, M., and Hennig, P. (2020a). Differentiable likelihoods for fast inversion of likelihood-free dynamical systems. In III, H. D. and Singh, A., editors,

Proceedings of the 37th International Conference on Machine Learning, volume 119, pages 5198–5208. PMLR.

Kersting, H., Sullivan, T., and Hennig, P. (2020b). Convergence rates of Gaussian ODE filters. *Statistics and Computing*, 30:1–26.

Kidger, P. (2021). *On Neural Differential Equations*. PhD thesis, University of Oxford.

Kohberger, R. C., Scavia, D., and Wilkinson, J. W. (1978). A method for parameter sensitivity analysis in differential equation models. *Water Resources Research*, 14(1):25–29.

Krämer, N., Bosch, N., Schmidt, J., and Hennig, P. (2022). Probabilistic ODE solutions in millions of dimensions. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 11634–11649. PMLR.

Krämer, N. and Hennig, P. (2020). Stable implementation of probabilistic ODE solvers. *Preprint, arXiv:2012.10106*.

Lazarus, A., Husmeier, D., and Papamarkou, T. (2018). Multiphase MCMC sampling for parameter inference in nonlinear ordinary differential equations. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84, pages 1252–1260. PMLR.

Ljung, L. (1979). Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control*, 24(1):36–50.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20:130–141.

Ma, Y., Dixit, V., Innes, M., Guo, X., and Rackauckas, C. (2021). A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions. *Preprint, arXiv:1812.01892*.

Macdonald, B., Higham, C., and Husmeier, D. (2015). Controversy in mechanistic modelling with Gaussian processes. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1539–1547. PMLR.

MacKay, D. J. C. (1992). A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472.

Nagumo, J., Arimoto, S., and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proceedings of the Institute of Radio Engineers*, 50(10):2061–2070.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, NY, USA, 2e edition.

Prague, M., Wittkop, L., Collin, A., Dutartre, D., Clairon, Q., Moireau, P., Thiébaut, R., and Hejblum, B. P. (2020). Multi-level modeling of early COVID-19 epidemic dynamics in French regions and estimation of the lockdown impact on infection rate. *Preprint, 10.1101/2020.04.21.20073536*.

Prince, P. J. and Dormand, J. R. (1981). High order embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 7(1):67–75.

Ramsay, J. O., Hooker, G., Campbell, D., and Cao, J. (2007). Parameter estimation for differential equations: A generalized smoothing approach. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 69(5):741–796.

Rocsoreanu, C., Georgescu, A., and Giurgiteanu, N. (2012). *The FitzHugh-Nagumo model: Bifurcation and Dynamics*, volume 10. Springer Science & Business Media.

Särkkä, S. and Svensson, L. (2023). *Bayesian Filtering and Smoothing*. Institute of Mathematical Stat. Cambridge University Press.

Schmidt, J., Krämer, N., and Hennig, P. (2021). A probabilistic state space model for joint inference from differential equations and data. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12374–12385. Curran Associates, Inc.

Schober, M., Duvenaud, D. K., and Hennig, P. (2014). Probabilistic ODE solvers with Runge-Kutta means. In *Advances in Neural Information Processing Systems*, pages 739–747. Curran Associates, Inc.

Schober, M., Särkkä, S., and Hennig, P. (2019). A probabilistic model for the numerical solution of initial value problems. *Statistics and Computing*, 29(1):99–122.

Skilling, J. (1992). *Bayesian Solution of Ordinary Differential Equations*, pages 23–37. Springer Netherlands.

Teymur, O., Zygalakis, K., and Calderhead, B. (2016). Probabilistic linear multistep methods. In *Advances in Neural Information Processing Systems*, pages 4321–4328. Curran Associates, Inc.

Tronarp, F., Bosch, N., and Hennig, P. (2022). Fenrir: Physics-enhanced regression for initial value problems. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings*

of the 39th International Conference on Machine Learning, volume 162, pages 21776–21794. PMLR.

Tronarp, F., Kersting, H., Särkkä, S., and Hennig, P. (2019). Probabilistic solutions to ordinary differential equations as non-linear Bayesian filtering: a new perspective. *Statistics and Computing*, 29:1297–1315.

Tronarp, F., Särkkä, S., and Hennig, P. (2021). Bayesian ODE solvers: the maximum a posteriori estimate. *Statistics and Computing*, 31(3):23.

Weber, F., Theers, S., Surmann, D., Ligges, U., and Weihs, C. (2018). Sensitivity analysis of ordinary differential equation models. Technical report, Technische Universität Dortmund.

Wenk, P., Gotovos, A., Bauer, S., Gorbach, N. S., Krause, A., and Buhmann, J. M. (2019). Fast Gaussian process based gradient matching for parameter identification in systems of nonlinear ODEs. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89, pages 1351–1360. PMLR.

Wong, S. W. K., Yang, S., and Kou, S. C. (2023). Magi: A package for inference of dynamic systems from noisy and sparse data via manifold-constrained gaussian processes. *Preprint, arXiv:2203.06066*.

Yang, S., Wong, S. W. K., and Kou, S. C. (2021). Inference of dynamic systems from noisy and sparse data via manifold-constrained Gaussian processes. *Proceedings of the National Academy of Sciences*, 118(15):e2020397118.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes.**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Yes. See "Computational Scaling" in Section 4 and Section 6.**
   (c) (Optional) Source code, with specification of all dependencies, including external libraries. **Yes. Link to source code is provided in Section 5.**

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Not Applicable. Lack of theoretical guarantees are discussed in the Conclusion.**
   (b) Complete proofs of all theoretical results. **NA.**
   (c) Clear explanations of any assumptions. **NA.**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **No. We only provide implementations of our algorithms. However, these are packaged as a Python library with detailed usage instructions, from which all results can be reproduced from the complete specification of the experiments we have provided in Sections 5 and D.**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes. See (a).**
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes.**
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **No. All computations were performed on a regular laptop.**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **Not Applicable. All data is simulated.**
   (b) The license information of the assets, if applicable. **NA.**
   (c) New assets either in the supplemental material or as a URL, if applicable. **NA.**
   (d) Information about consent from data providers/curators. **NA.**
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **NA.**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. **NA.**
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **NA.**
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **NA.**

A HIGHER ORDER ODE SYSTEMS

For a multi-variable function $\boldsymbol{x}(t) = \big(x_1(t), \dots, x_d(t)\big)$, an arbitrary-order ODE-IVP can be written as

$$\boldsymbol{W}\boldsymbol{X}(t) = \boldsymbol{f}(\boldsymbol{X}(t), t), \qquad \boldsymbol{X}(0) = \boldsymbol{v}, \qquad t \in [0, T], \tag{34}$$

where $\boldsymbol{X}(t) = (\boldsymbol{X}^{(1)}(t), \dots, \boldsymbol{X}^{(d)}(t))$,

$$\boldsymbol{X}^{(k)}(t) = \left(x_k(t), \dot{x}_k(t), \dots, \frac{\mathrm{d}^{q_k-1}}{\mathrm{d}t^{q_k-1}} x_k(t)\right) \tag{35}$$

contains $x_k(t)$ and its first $q_k - 1$ derivatives, $\boldsymbol{W}$ is a coefficient matrix of size $r \times q$, where $q = \sum_{k=1}^{d} q_k$, and $\boldsymbol{f} : \mathbb{R}^q \to \mathbb{R}^r$ is the (typically nonlinear) ODE function. For the usual ODE-IVP formulation of $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{X}(t) = \boldsymbol{f}(\boldsymbol{X}(t), t)$, we have $\boldsymbol{W} = \mathrm{diag}(\boldsymbol{S}_{q_1}, \dots, \boldsymbol{S}_{q_d})$, where $\boldsymbol{S}_{q_k} = [\boldsymbol{0}_{(q_k-1)\times 1} \mid \boldsymbol{I}_{(q_k-1)\times(q_k-1)}]$ for $k = 1, \dots, d$. The Bayesian filtering model (3) in the data-free setting then becomes

$$\begin{aligned} \boldsymbol{X}_{n+1} \mid \boldsymbol{X}_n &\sim \mathrm{Normal}(\boldsymbol{Q}\boldsymbol{X}_n, \boldsymbol{R}) \\ \boldsymbol{Z}_n &\overset{\mathrm{ind}}{\sim} \mathrm{Normal}(\boldsymbol{W}\boldsymbol{X}_n - \boldsymbol{f}(\boldsymbol{X}_n, t_n), \boldsymbol{V}), \end{aligned} \tag{36}$$

with subsequent linearizations, data-driven extensions, etc., following straighforwardly from (36).

B GAUSSIAN MARKOV PROCESS PRIOR

For the multi-variable function $\boldsymbol{x}(t) = (x_1(t), \dots, x_d(t))$, DALTON uses a simple and effective prior proposed by Schober et al. (2019); namely, that each $x_k(t)$ follows an independent $(p_k - 1)$-times integrated Brownian motion (IBM),

$$\frac{\mathrm{d}^{p_k-1}}{\mathrm{d}t^{p_k-1}} x_k(t) = \sigma_k B_k(t), \tag{37}$$

where $B_1(t), \dots, B_d(t)$ are independent Wiener processes. This prior formulation gives $x_k(t)$ $p_k - 1$ continuous derivatives. It is therefore desirable to set $p_k > q_k$ to have a sufficiently smooth solution process. In our experiments we have set $p_k = q_k + 1$, and redefined the ODE system (34) to be of order $q_k = p_k$ by padding $\boldsymbol{W}$ with zeros.

The IBM prior (37) defines $\boldsymbol{X}^{(k)}(t)$ in (35) as a continuous Gaussian Markov process,

$$\boldsymbol{X}^{(k)}(t + \Delta t) \mid \boldsymbol{X}^{(k)}(t) \sim \mathrm{Normal}(\boldsymbol{Q}^{(k)}\boldsymbol{X}^{(k)}(t), \boldsymbol{R}^{(k)}), \tag{38}$$

with the elements of matrices $\boldsymbol{Q}^{(k)}$ and $\boldsymbol{R}^{(k)}$ given by

$$Q^{(k)}_{ij} = \mathbb{I}(i \le j) \cdot \frac{(\Delta t)^{j-i}}{(j-i)!}, \qquad R^{(k)}_{ij} = \sigma_k^2 \cdot \frac{(\Delta t)^{2p-1-i-j}}{(2p-1-i-j)(p-1-i)!(p-1-j)!}. \tag{39}$$

The full matrices in the Bayesian filtering model (36) are $\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{Q}^{(1)}, \dots, \boldsymbol{Q}^{(d)})$ and $\boldsymbol{R} = \mathrm{diag}(\boldsymbol{R}^{(1)}, \dots, \boldsymbol{R}^{(d)})$. In the context of parameter learning, the parameters $\boldsymbol{\eta}$ of the prior process are the IBM scale parameters, $\boldsymbol{\eta} = (\sigma_1, \dots, \sigma_d)$.

C EFFICIENT AND ACCURATE ODE LINEARIZATION

In the experiments of Section 5, we employ a modified first-order linearization of $\boldsymbol{f}(\boldsymbol{X}_n, t_n)$ in (34) proposed by Tronarp et al. (2019). That is, the first-order Taylor expansion of $\boldsymbol{f}(\boldsymbol{X}_n, t_n$ is

$$\boldsymbol{f}(\boldsymbol{X}_n, t_n) \approx \boldsymbol{f}(\boldsymbol{\mu}_{n|n-1}, t_n) + \boldsymbol{J}_n(\boldsymbol{X}_n - \boldsymbol{\mu}_{n|n-1}) \tag{40}$$

where $\boldsymbol{\mu}_{n|n-1}$ is the mean of the predictive distribution of the linearized working model (5) or (14), depending on whether the linearization is data-free or data-dependent, and $\boldsymbol{J}_n = \frac{\partial}{\partial \boldsymbol{X}_n}\boldsymbol{f}(\boldsymbol{\mu}_{n|n-1}, t_n)$. The modified linearization is then

$$\boldsymbol{a}_n = -\boldsymbol{f}(\boldsymbol{\mu}_{n|n-1}, t_n) + \boldsymbol{J}_n\boldsymbol{\mu}_{n|n-1}, \quad \boldsymbol{B}_n = \mathrm{diag}\left(\frac{\partial}{\partial \boldsymbol{X}_n^{(1)}}\boldsymbol{f}(\boldsymbol{\mu}_{n|n-1}, t_n), \dots, \frac{\partial}{\partial \boldsymbol{X}_n^{(d)}}\boldsymbol{f}(\boldsymbol{\mu}_{n|n-1}, t_n)\right), \quad \boldsymbol{V}_n = \boldsymbol{0}, \tag{41}$$

where $\boldsymbol{X}_n = (\boldsymbol{X}_n^{(1)}, \ldots, \boldsymbol{X}_n^{(d)})$. In other words, $\boldsymbol{B}_n$ keeps only the block diagonal entries of $\boldsymbol{J}_n$. With corresponding block diagonal matrices $\boldsymbol{Q}_{\boldsymbol{\eta}}$ and $\boldsymbol{R}_{\boldsymbol{\eta}}$, the Kalman recursions underlying the Fenrir and DALTON algorithms can be performed blockwise (Krämer et al., 2022). Thus for $q_k \equiv p$, the scaling of these algorithms drops from $\mathcal{O}(N(dp)^3)$ to $\mathcal{O}(Ndp^3)$, a substantial improvement for low-order many-variable ODE systems.

D IMPLEMENTATION DETAILS FOR PARAMETER LEARNING METHODS

In the experiments of Section 5, We take the measurement model parameters $\boldsymbol{\phi}$ to be known, such that the learning problem is only for the ODE model parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_D)$ and the tuning parameter $\boldsymbol{\eta} = (\sigma_1, \ldots, \sigma_d)$ of the IBM prior in Appendix B. We use a flat prior on $\boldsymbol{\eta}$ and independent Normal$(0, 10^2)$ priors on either θ_r or $\log \theta_r$, $r = 1, \ldots, D$, depending on whether θ_r is unbounded or $\theta_r > 0$.

For the unknown model parameters $\boldsymbol{\theta}$, we use the Laplace posterior approximation

$$\boldsymbol{\theta} \mid \boldsymbol{Y}_{0:M} \approx \text{Normal}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{V}}_{\boldsymbol{\theta}}), \tag{42}$$

where $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\eta}}) = \arg\max_{(\boldsymbol{\theta},\boldsymbol{\eta})} \log p(\boldsymbol{\theta}, \boldsymbol{\eta} \mid \boldsymbol{Y}_{0:M})$ is obtained using the Newton-CG optimization algorithm (Nocedal and Wright, 2006) as implemented in the Python **JAXopt** library (Blondel et al., 2022). As for $\hat{\boldsymbol{V}}_{\boldsymbol{\theta}}$, we estimate it as $\hat{\boldsymbol{V}}_{\boldsymbol{\theta}} = -\big[\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'} \log p(\hat{\boldsymbol{\Theta}} \mid \boldsymbol{Y}_{0:M})\big]^{-1}$, i.e., with the Hessian taken only with respect to $\boldsymbol{\theta}$ and not $\boldsymbol{\eta}$. This approach was used in Tronarp et al. (2022), and found here to produce slightly better results than when uncertainty is propagated through the prior process tuning parameters as well.

The MAGI algorithm as implemented in Wong et al. (2023) uses a slighly different prior specification, i.e., with a Matérn covariance function with smoothing parameter $\nu = 2.01$, which ensures twice-continuous differentiability of the prior on $\boldsymbol{x}(t)$, and scale parameters σ_k corresponding to the maximum *a posteriori* (MAP) estimates of the Gaussian process regression conditioning only on the observed data, which is analytically tractable since this implementation of MAGI is restricted to Gaussian measurement models. The implementation then uses a hybrid Monte Carlo (HMC) algorithm to sample from $p(\boldsymbol{\theta}, \boldsymbol{X}_{0:N} \mid \boldsymbol{Y}_{0:M})$.

Our implementation of the Chkbretii MCMC algorithm starts by fitting a DALTON parameter learning algorithm, which we do not factor into the timings presented in Table 1. From this preliminary inference run we fix the Gaussian process tuning parameters at $\hat{\boldsymbol{\eta}}$. The Chkbretii algorithm still requires the specification of a proposal distribution $q(\boldsymbol{\theta} \mid \boldsymbol{\theta}')$. We set this as $\boldsymbol{\theta} \sim \text{Normal}(\boldsymbol{\theta}', \tau \cdot \hat{\boldsymbol{V}}_{\boldsymbol{\theta}})$, where $\hat{\boldsymbol{V}}_{\boldsymbol{\theta}}$ is obtained from the DALTON algorithm and τ is tuned during the burning stage to obtain an overall acceptance rate of 25% (also not counted in the timings of Table 1).

The MCMC algorithm used to explore the full DALTON and RKDP posteriors is a random walk with the same proposal distribution as for the Chkbretii algorithm above.

E DALTON ALGORITHMS

For Algorithms 1 and 2, we use the extension discussed in Appendix A to generalize to arbitrary-order ODEs. The `linearize`() function (e.g., Line 9 of Algorithm 1) performs the modified first-order linearization detailed in Appendix C, and `normal_logpdf`() on Line 11 is the log-pdf of a multivariate normal.

E.1 Gaussian Measurements

Algorithm 1 DALTON probabilistic ODE likelihood approximation for Gaussian measurements.

1: **procedure** DALTON($\boldsymbol{W}_{\boldsymbol{\theta}}, \boldsymbol{f}(\boldsymbol{X}, t, \boldsymbol{\theta}), \boldsymbol{v}_{\boldsymbol{\theta}}, \boldsymbol{Q}_{\boldsymbol{\eta}}, \boldsymbol{R}_{\boldsymbol{\eta}}, \boldsymbol{Y}_{0:M}, \boldsymbol{D}^{\boldsymbol{\phi}}_{0:M}, \boldsymbol{\Omega}^{\boldsymbol{\phi}}_{0:M}$)
2: $\quad\boldsymbol{\mu}_{0|0}, \Sigma_{0|0} \leftarrow \boldsymbol{v}, \boldsymbol{0}$ ▷ Initialization
3: $\quad\boldsymbol{Z}_{0:N} \leftarrow \boldsymbol{0}$
4: $\quad\ell_z, \ell_{yz} \leftarrow 0, 0$
5: $\quad i \leftarrow 0$ ▷ Used to map t_n to t'_i
6: ▷ Lines 7-12 compute $\log p(\boldsymbol{Z}_{0:N} = \boldsymbol{0} \mid \boldsymbol{\Theta})$
7: $\quad$**for** $n = 1 : N$ **do**
8: $\quad\quad\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1} \leftarrow$ `kalman_predict`$(\boldsymbol{\mu}_{n-1|n-1}, \boldsymbol{\Sigma}_{n-1|n-1}, \boldsymbol{0}, \boldsymbol{Q}_{\boldsymbol{\eta}}, \boldsymbol{R}_{\boldsymbol{\eta}})$
9: $\quad\quad\boldsymbol{a}_n, \boldsymbol{B}_n, \boldsymbol{V}_n \leftarrow$ `linearize`$(\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}, \boldsymbol{W}_{\boldsymbol{\theta}}, \boldsymbol{f}(\boldsymbol{X}, t_n, \boldsymbol{\theta}))$
10: $\quad\quad\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n \leftarrow$ `kalman_forecast`$(\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}, \boldsymbol{a}_n, \boldsymbol{W}_{\boldsymbol{\theta}} + \boldsymbol{B}_n, \boldsymbol{V}_n)$
11: $\quad\quad\ell_z \leftarrow \ell_z +$ `normal_logpdf`$(\boldsymbol{Z}_n = \boldsymbol{0}; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$
12: $\quad\quad\boldsymbol{\mu}_{n|n}, \boldsymbol{\Sigma}_{n|n} \leftarrow$ `kalman_update`$(\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}, \boldsymbol{Z}_n, \boldsymbol{a}_n, \boldsymbol{W}_{\boldsymbol{\theta}} + \boldsymbol{B}_n, \boldsymbol{V}_n)$
13: ▷ Lines 14-24 compute $\log p(\boldsymbol{Y}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0} \mid \boldsymbol{\Theta})$
14: $\quad\ell_{yz} \leftarrow$ `normal_logpdf`$(\boldsymbol{Y}_0; \boldsymbol{D}^{\boldsymbol{\phi}}_0 \boldsymbol{v}_{\boldsymbol{\theta}}, \boldsymbol{\Omega}^{\boldsymbol{\phi}}_0)$
15: $\quad$**for** $n = 1 : N$ **do**
16: $\quad\quad\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1} \leftarrow$ `kalman_predict`$(\boldsymbol{\mu}_{n-1|n-1}, \boldsymbol{\Sigma}_{n-1|n-1}, \boldsymbol{0}, \boldsymbol{Q}_{\boldsymbol{\eta}}, \boldsymbol{R}_{\boldsymbol{\eta}})$
17: $\quad\quad\boldsymbol{a}_n, \boldsymbol{B}_n, \boldsymbol{V}_n \leftarrow$ `linearize`$(\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}, \boldsymbol{W}_{\boldsymbol{\theta}}, \boldsymbol{f}(\boldsymbol{X}, t_n, \boldsymbol{\theta}))$
18: $\quad\quad$**if** $t_n = t_{n(i)}$ **then**
19: $\quad\quad\quad\boldsymbol{Z}_n \leftarrow \begin{bmatrix} \boldsymbol{Z}_n \\ \boldsymbol{Y}_i \end{bmatrix}, \qquad \boldsymbol{W}_{\boldsymbol{\theta}} \leftarrow \begin{bmatrix} \boldsymbol{W}_{\boldsymbol{\theta}} \\ \boldsymbol{D}^{\boldsymbol{\phi}}_i \end{bmatrix}, \qquad \boldsymbol{B}_n \leftarrow \begin{bmatrix} \boldsymbol{B}_n \\ \boldsymbol{0} \end{bmatrix}$
20: $\quad\quad\quad\boldsymbol{a}_n \leftarrow \begin{bmatrix} \boldsymbol{a}_n \\ \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{V}_n \leftarrow \begin{bmatrix} \boldsymbol{V}_n & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Omega}^{\boldsymbol{\phi}}_i \end{bmatrix}$
21: $\quad\quad\quad i \leftarrow i + 1$
22: $\quad\quad\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n \leftarrow$ `kalman_forecast`$(\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}, \boldsymbol{a}_n, \boldsymbol{W}_{\boldsymbol{\theta}} + \boldsymbol{B}_n, \boldsymbol{V}_n)$
23: $\quad\quad\ell_{yz} \leftarrow \ell_{yz} +$ `normal_logpdf`$(\boldsymbol{Z}_n; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$
24: $\quad\quad\boldsymbol{\mu}_{n|n}, \boldsymbol{\Sigma}_{n|n} \leftarrow$ `kalman_update`$(\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}, \boldsymbol{Z}_n, \boldsymbol{a}_n, \boldsymbol{W}_{\boldsymbol{\theta}} + \boldsymbol{B}_n, \boldsymbol{V}_n)$
25:
26: $\quad$**return** $\ell_{yz} - \ell_z$ ▷ Estimate of $\log p(\boldsymbol{Y}_{0:M} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}, \boldsymbol{\Theta})$

E.2 Non-Gaussian Measurements

Recall in Section 3.2 that we denote $\boldsymbol{s}_{n(i)}$ and $\boldsymbol{u}_{n(i)}$ to be the observed and unobserved components of $\boldsymbol{x}(t)$ at time $t = t'_i$. In order to present Algorithm 2, we introduce the notation $\boldsymbol{s}_{n(i)} = \boldsymbol{D}_i \boldsymbol{X}_{n(i)}$, where $\boldsymbol{D}_i$ is a mask matrix of zeros and ones, so as to more naturally relate to the Gaussian version of DALTON in Algorithm 1.

Algorithm 2 DALTON probabilistic ODE likelihood approximation for non-Gaussian measurements.

procedure DALTON($\boldsymbol{W_\theta}, \boldsymbol{f}(\boldsymbol{X}, t, \boldsymbol{\theta}), \boldsymbol{v_\theta}, \boldsymbol{Q_\eta}, \boldsymbol{R_\eta}, \boldsymbol{Y}_{0:M}, \boldsymbol{\phi}, \boldsymbol{g}_{0:M}(\boldsymbol{Y}, \boldsymbol{s}, \boldsymbol{\phi})$)
  $\boldsymbol{\mu}_{0|0}, \Sigma_{0|0} \leftarrow \boldsymbol{v}, \boldsymbol{0}$ ▷ Initialization
  $\boldsymbol{Z}_{0:N} \leftarrow \boldsymbol{0}$
  $\ell_{xz}, \ell_{xyz}\ell_y \leftarrow 0, 0, 0$
  $i \leftarrow 0$ ▷ Used to map t_n to t'_i
  ▷ Lines 7-24 compute $\log p(\boldsymbol{X}_{0:N} \mid \hat{\boldsymbol{Y}}_{0:M}, \boldsymbol{Z}_{0:N} = \boldsymbol{0}, \boldsymbol{\Theta})$
  for $n = 1 : N$ **do**
    $\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1} \leftarrow \texttt{kalman\_predict}(\boldsymbol{\mu}_{n-1|n-1}, \boldsymbol{\Sigma}_{n-1|n-1}, \boldsymbol{0}, \boldsymbol{Q_\eta}, \boldsymbol{R_\eta})$
    $\boldsymbol{a}_n, \boldsymbol{B}_n, \boldsymbol{V}_n \leftarrow \texttt{linearize}(\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}, \boldsymbol{W_\theta}, \boldsymbol{f}(\boldsymbol{X}, t_n, \boldsymbol{\theta}))$
    if $t_n = t_{n(i)}$ **then**
      $\boldsymbol{g}_{(1)} \leftarrow \frac{\partial}{\partial \boldsymbol{s}_{n(i)}} g_i(\boldsymbol{Y}_i, \boldsymbol{D}_i \boldsymbol{\mu}_{n|n-1}, \boldsymbol{\phi})$
      $\boldsymbol{g}_{(2)} \leftarrow \frac{\partial^2}{\partial \boldsymbol{s}_{n(i)} \partial \boldsymbol{s}'_{n(i)}} g_i(\boldsymbol{Y}_i, \boldsymbol{D}_i \boldsymbol{\mu}_{n|n-1}, \boldsymbol{\phi})$
      $\hat{\boldsymbol{Y}}_i \leftarrow \boldsymbol{D}_i \boldsymbol{\mu}_{n|n-1} - \boldsymbol{g}_{(2)}^{-1} \boldsymbol{g}_{(1)}$ ▷ Compute pseudo-observations
      $\boldsymbol{Z}_n \leftarrow \begin{bmatrix} \boldsymbol{Z}_n \\ \hat{\boldsymbol{Y}}_i \end{bmatrix}, \qquad \boldsymbol{W_\theta} \leftarrow \begin{bmatrix} \boldsymbol{W_\theta} \\ \boldsymbol{D}_i \end{bmatrix}, \qquad \boldsymbol{B}_n \leftarrow \begin{bmatrix} \boldsymbol{B}_n \\ \boldsymbol{0} \end{bmatrix}$
      $\boldsymbol{a}_n \leftarrow \begin{bmatrix} \boldsymbol{a}_n \\ \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{V}_n \leftarrow \begin{bmatrix} \boldsymbol{V}_n & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{g}_{(2)}^{-1} \end{bmatrix}$
      $i \leftarrow i + 1$
    $\boldsymbol{\mu}_{n|n}, \boldsymbol{\Sigma}_{n|n} \leftarrow \texttt{kalman\_update}(\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}, \boldsymbol{Z}_n, \boldsymbol{a}_n, \boldsymbol{W_\theta} + \boldsymbol{B}_n, \boldsymbol{V}_n)$
  for $n = N - 1 : 1$ **do**
    $\boldsymbol{\mu}_{n|N}, \boldsymbol{\Sigma}_{n|N} \leftarrow \texttt{kalman\_smooth}(\boldsymbol{\mu}_{n+1|N}, \boldsymbol{\Sigma}_{n+1|N}, \boldsymbol{\mu}_{n|n}, \boldsymbol{\Sigma}_{n|n}, \boldsymbol{\mu}_{n+1|n}, \boldsymbol{\Sigma}_{n+1|n}, \boldsymbol{Q_\eta})$
    $\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n \leftarrow \texttt{kalman\_sample}(\boldsymbol{\mu}_{n+1|N}, \boldsymbol{\mu}_{n|n}, \boldsymbol{\Sigma}_{n|n}, \boldsymbol{\mu}_{n+1|n}, \boldsymbol{\Sigma}_{n+1|n}, \boldsymbol{Q_\eta})$
    $\ell_{xyz} \leftarrow \ell_{xyz} + \texttt{normal\_logpdf}(\boldsymbol{\mu}_{n|N}; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$
  $\ell_{xyz} \leftarrow \ell_{xyz} + \texttt{normal\_logpdf}(\boldsymbol{\mu}_{N|N}; \boldsymbol{\mu}_{N|N}, \boldsymbol{\Sigma}_{N|N})$
  ▷ Lines 26-35 compute $\log p(\boldsymbol{X}_{0:N} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}, \boldsymbol{\Theta})$
  $\boldsymbol{Z}_{0:N} \leftarrow \boldsymbol{0}$ ▷ Reset $\boldsymbol{Z}_{0:N} = \boldsymbol{0}$
  $\boldsymbol{\mu}'_{0:N|N} \leftarrow \boldsymbol{\mu}_{0:N|N}$ ▷ Store $\boldsymbol{\mu}_{0:N|N} = E[\boldsymbol{X}_{0:N} \mid \boldsymbol{Z}_{0:N}, \hat{\boldsymbol{Y}}_{0:N}]$ for later
  for $n = 1 : N$ **do**
    $\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1} \leftarrow \texttt{kalman\_predict}(\boldsymbol{\mu}_{n-1|n-1}, \boldsymbol{\Sigma}_{n-1|n-1}, \boldsymbol{0}, \boldsymbol{Q_\eta}, \boldsymbol{R_\eta})$
    $\boldsymbol{a}_n, \boldsymbol{B}_n, \boldsymbol{V}_n \leftarrow \texttt{linearize}(\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}, \boldsymbol{W_\theta}, \boldsymbol{f}(\boldsymbol{X}, t_n, \boldsymbol{\theta}))$
    $\boldsymbol{\mu}_{n|n}, \boldsymbol{\Sigma}_{n|n} \leftarrow \texttt{kalman\_update}(\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}, \boldsymbol{Z}_n, \boldsymbol{a}_n, \boldsymbol{W_\theta} + \boldsymbol{B}_n, \boldsymbol{V}_n)$
  for $n = N - 1 : 1$ **do**
    $\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n \leftarrow \texttt{kalman\_sample}(\boldsymbol{\mu}'_{n+1|N}, \boldsymbol{\mu}_{n|n}, \boldsymbol{\Sigma}_{n|n}, \boldsymbol{\mu}_{n+1|n}, \boldsymbol{\Sigma}_{n+1|n}, \boldsymbol{Q_\eta})$
    $\ell_{xz} \leftarrow \ell_{xz} + \texttt{normal\_logpdf}(\boldsymbol{\mu}'_{n|N}; \boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$
  $\ell_{xz} \leftarrow \ell_{xz} + \texttt{normal\_logpdf}(\boldsymbol{\mu}'_{N|N}; \boldsymbol{\mu}_{N|N}, \boldsymbol{\Sigma}_{N|N})$
  ▷ Lines 37-39 compute $\log p(\boldsymbol{Y}_{0:M} \mid \boldsymbol{X}_{0:N}, \boldsymbol{\phi})$
  for $i = 0 : M$ **do**
    $\ell_y \leftarrow \ell_y - g_i(\boldsymbol{Y}_i, \boldsymbol{\mu}'_{n(i)|N}, \boldsymbol{\phi})$

  return $\ell_{xz} - \ell_y - \ell_{xyz}$ ▷ Estimate of $\log p(\boldsymbol{Y}_{0:M} \mid \boldsymbol{Z}_{0:N} = \boldsymbol{0}, \boldsymbol{\Theta})$

F KALMAN FUNCTIONS

The Kalman recursions used in Algorithms 1 and 2 are formulated in terms of the general Gaussian state space model

$$\begin{aligned} \boldsymbol{X}_n &= \boldsymbol{Q}_n \boldsymbol{X}_{n-1} + \boldsymbol{c}_n + \boldsymbol{R}_n^{1/2} \boldsymbol{\epsilon}_n \\ \boldsymbol{Z}_n &= \boldsymbol{W}_n \boldsymbol{X}_n + \boldsymbol{a}_n + \boldsymbol{V}_n^{1/2} \boldsymbol{\eta}_n \end{aligned}, \qquad \boldsymbol{\epsilon}_n, \boldsymbol{\eta}_n \overset{\text{ind}}{\sim} \text{Normal}(\boldsymbol{0}, \boldsymbol{I}). \tag{43}$$

Throughout, we use the notation $\boldsymbol{\mu}_{n|m} = E[\boldsymbol{X}_n \mid \boldsymbol{Z}_{0:m}]$ and $\boldsymbol{\Sigma}_{n|m} = \text{var}(\boldsymbol{X}_n \mid \boldsymbol{Z}_{0:m})$.

Algorithm 3 Standard Kalman filtering and smoothing recursions.

procedure KALMAN_PREDICT($\boldsymbol{\mu}_{n-1|n-1}, \boldsymbol{\Sigma}_{n-1|n-1}, \boldsymbol{c}_n, \boldsymbol{Q}_n, \boldsymbol{R}_n$)
  $\boldsymbol{\mu}_{n|n-1} \leftarrow \boldsymbol{Q}_n \boldsymbol{\mu}_{n-1|n-1} + \boldsymbol{c}_n$
  $\boldsymbol{\Sigma}_{n|n-1} \leftarrow \boldsymbol{Q}_n \boldsymbol{\Sigma}_{n-1|n-1} \boldsymbol{Q}_n' + \boldsymbol{R}_n$
  return $\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}$
procedure KALMAN_UPDATE($\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}, \boldsymbol{Z}_n, \boldsymbol{a}_n, \boldsymbol{W}_n, \boldsymbol{V}_n$)
  $\boldsymbol{A}_n \leftarrow \boldsymbol{\Sigma}_{n|n-1} \boldsymbol{W}_n' [\boldsymbol{W}_n \boldsymbol{\Sigma}_{n|n-1} \boldsymbol{W}_n' + \boldsymbol{V}_n]^{-1}$
  $\boldsymbol{\mu}_{n|n} \leftarrow \boldsymbol{\mu}_{n|n-1} + \boldsymbol{A}_n (\boldsymbol{Z}_n - \boldsymbol{W}_n \boldsymbol{\mu}_{n|n-1} - \boldsymbol{a}_n)$
  $\boldsymbol{\Sigma}_{n|n} \leftarrow \boldsymbol{\Sigma}_{n|n-1} - \boldsymbol{A}_n \boldsymbol{W}_n \boldsymbol{\Sigma}_{n|n-1}$
  return $\boldsymbol{\mu}_{n|n}, \boldsymbol{\Sigma}_{n|n}$
procedure KALMAN_SMOOTH($\boldsymbol{\mu}_{n+1|N}, \boldsymbol{\Sigma}_{n+1|N}, \boldsymbol{\mu}_{n|n}, \boldsymbol{\Sigma}_{n|n}, \boldsymbol{\mu}_{n+1|n}, \boldsymbol{\Sigma}_{n+1|n}, \boldsymbol{Q}_{n+1}$)
  $\boldsymbol{A}_n \leftarrow \boldsymbol{\Sigma}_{n|n} \boldsymbol{Q}_{n+1}' \boldsymbol{\Sigma}_{n+1|n}^{-1}$
  $\boldsymbol{\mu}_{n|N} \leftarrow \boldsymbol{\mu}_{n|n} + \boldsymbol{A}_n (\boldsymbol{\mu}_{n+1|N} - \boldsymbol{\mu}_{n+1|n})$
  $\boldsymbol{\Sigma}_{n|N} \leftarrow \boldsymbol{\Sigma}_{n|n} + \boldsymbol{A}_n (\boldsymbol{\Sigma}_{n+1|N} - \boldsymbol{\Sigma}_{n+1|n}) \boldsymbol{A}_n'$
  return $\boldsymbol{\mu}_{n|N}, \boldsymbol{\Sigma}_{n|N}$
procedure KALMAN_SAMPLE($\boldsymbol{x}_{n+1}, \boldsymbol{\mu}_{n|n}, \boldsymbol{\Sigma}_{n|n}, \boldsymbol{\mu}_{n+1|n}, \boldsymbol{\Sigma}_{n+1|n}, \boldsymbol{Q}_{n+1}$)
  $\boldsymbol{A}_n \leftarrow \boldsymbol{\Sigma}_{n|n} \boldsymbol{Q}_{n+1}' \boldsymbol{\Sigma}_{n+1|n}^{-1}$
  $\tilde{\boldsymbol{\mu}}_{n|N} \leftarrow \boldsymbol{\mu}_{n|n} + \boldsymbol{A}_n (\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_{n+1|n})$ ▷ $\tilde{\boldsymbol{\mu}}_{n|N} = E[\boldsymbol{X}_n \mid \boldsymbol{X}_{n+1:N}, \boldsymbol{Z}_{0:N}]$
  $\tilde{\boldsymbol{\Sigma}}_{n|N} \leftarrow \boldsymbol{\Sigma}_{n|n} - \boldsymbol{A}_n \boldsymbol{Q}_{n+1} \boldsymbol{\Sigma}_{n|n}$ ▷ $\tilde{\boldsymbol{\Sigma}}_{n|N} = \text{var}[\boldsymbol{X}_n \mid \boldsymbol{X}_{n+1:N}, \boldsymbol{Z}_{0:N}]$
  return $\tilde{\boldsymbol{\mu}}_{n|N}, \tilde{\boldsymbol{\Sigma}}_{n|N}$
procedure KALMAN_FORECAST($\boldsymbol{\mu}_{n|n-1}, \boldsymbol{\Sigma}_{n|n-1}, \boldsymbol{a}_n, \boldsymbol{W}_n, \boldsymbol{V}_n$)
  $\boldsymbol{\lambda}_n \leftarrow \boldsymbol{W}_n \boldsymbol{\mu}_{n|n-1} + \boldsymbol{a}_n$ ▷ $\boldsymbol{\lambda}_{n|n-1} = E[\boldsymbol{Z}_n \mid \boldsymbol{Z}_{0:n-1}]$
  $\boldsymbol{\Omega}_{n|n-1} \leftarrow \boldsymbol{W}_n \boldsymbol{\mu}_{n|n-1} \boldsymbol{W}_n' + \boldsymbol{V}_n$ ▷ $\boldsymbol{\Omega}_{n|n-1} = \text{var}[\boldsymbol{Z}_n \mid \boldsymbol{Z}_{0:n-1}]$
  return $\boldsymbol{\lambda}_{n|n-1}, \boldsymbol{\Omega}_{n|n-1}$

G ADDITIONAL FIGURES

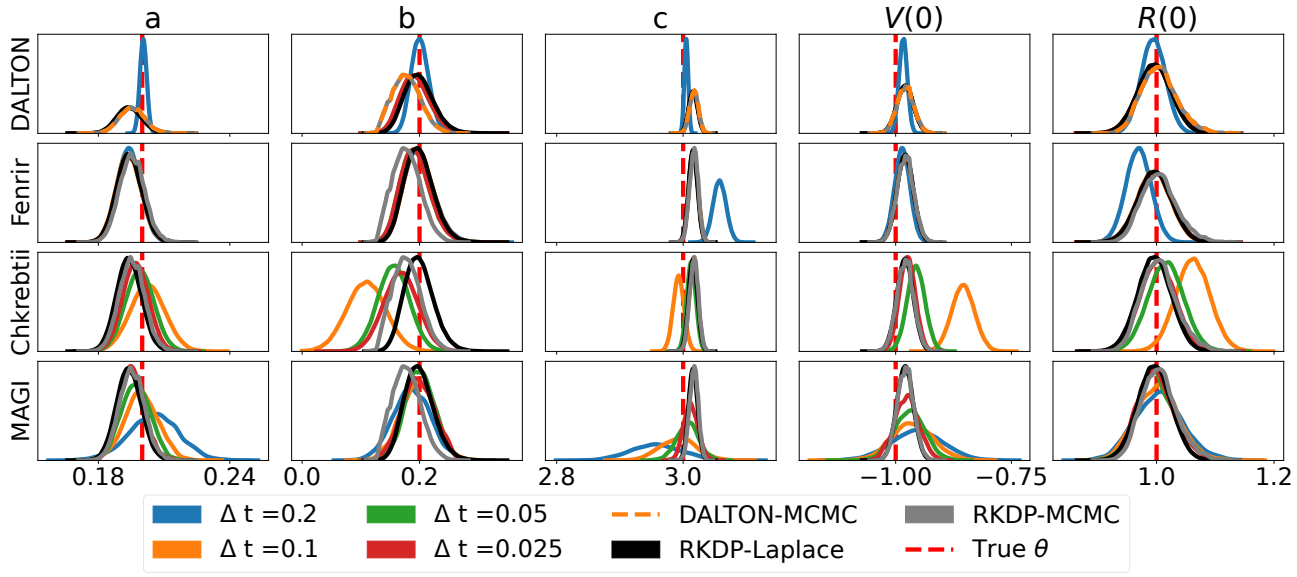

Figure 9: Parameter posteriors for the FN model using various ODE solvers. The two additional curves relative to Figure 2 are: (i) DALTON-MCMC: the exact DALTON posteriors at $\Delta t = 0.1$ (top row); (ii) RKDP-MCMC: the exact RKDP posteriors (all subplots). Both sets of exact posteriors are obtained using the MCMC algorithm described in Section D.

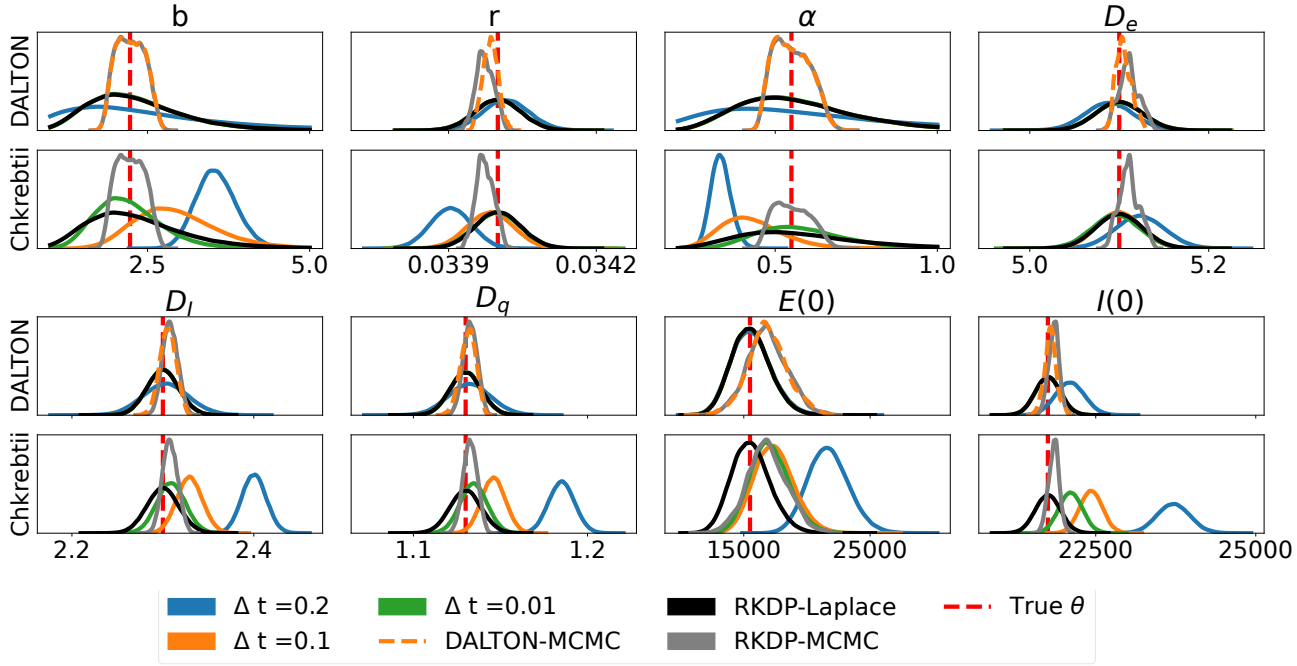

Figure 10: Parameter posteriors for the SEIRAH model using various ODE solvers. The two additional curves relative to Figure 3 are: (i) DALTON-MCMC: the exact DALTON posteriors at $\Delta t = 0.1$ (rows 1 and 3); (ii) RKDP-MCMC: the exact RKDP posteriors (all subplots). Both sets of exact posteriors are obtained using the MCMC algorithm described in Section D.

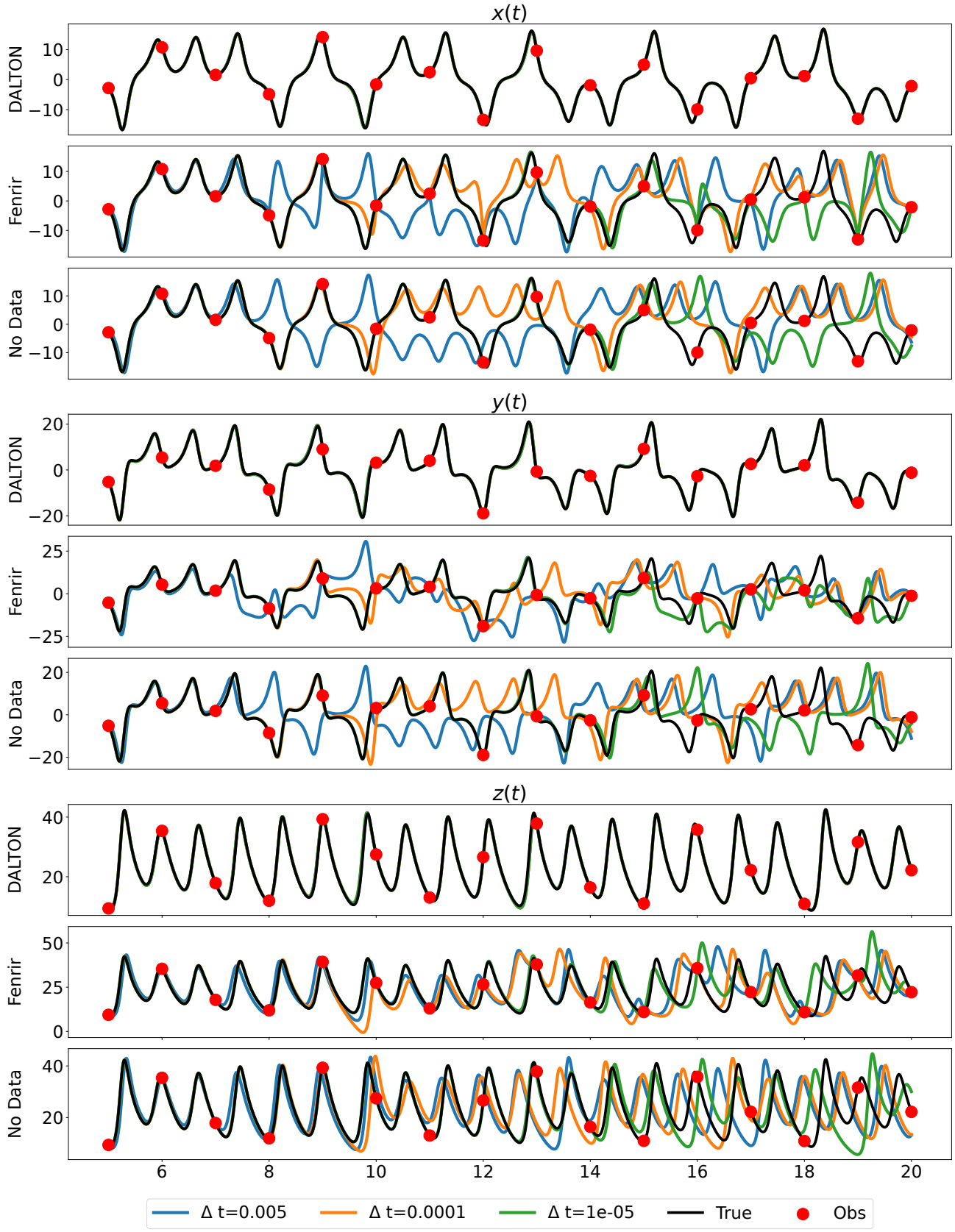

Figure 11: ODE solution for the Lorenz63 model calculated using various solvers. Extended version of Figure 5.

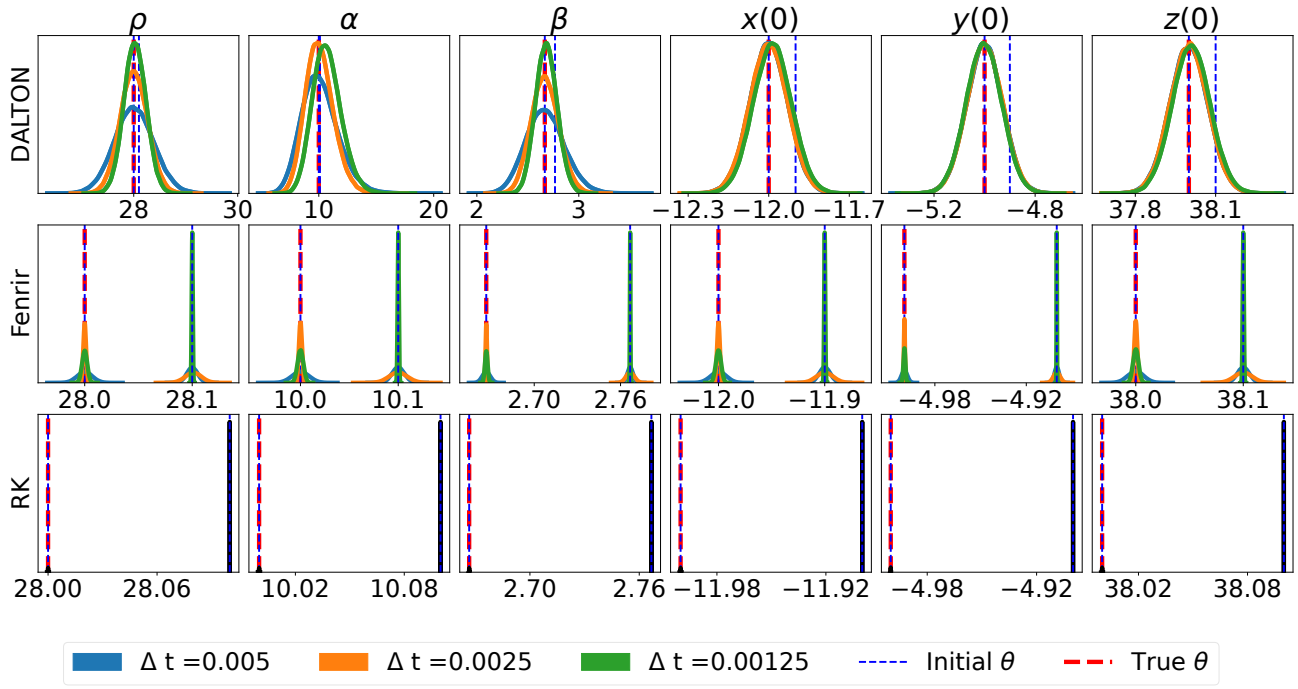

Figure 12: Parameter posteriors for the Lorenz63 model using various ODE solvers at two initial parameter values. Extended version of Figure 6.

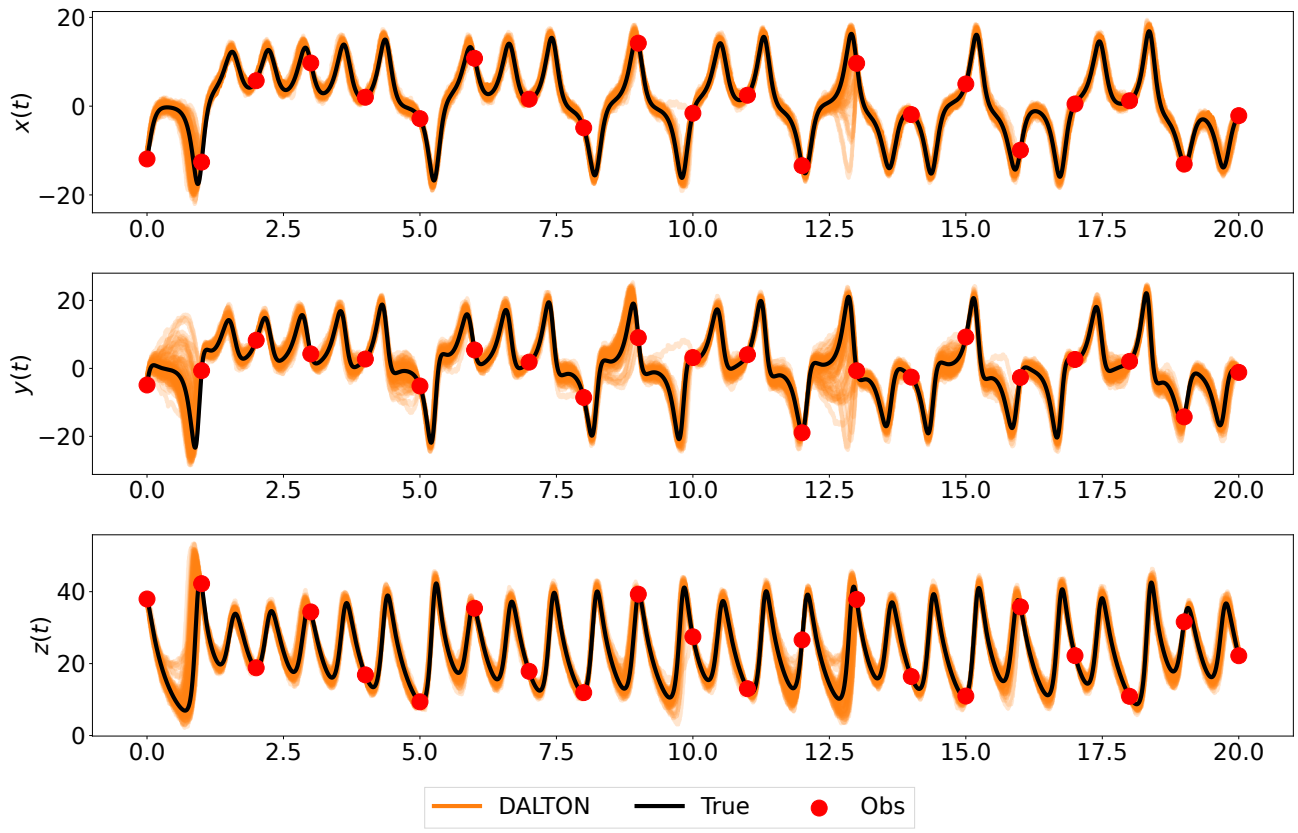

Figure 13: ODE posteriors for the Lorenz63 model using DALTON. Extended version of Figure 8.