ODIN: ODE-Informed Regression for Parameter and State Inference in Time-Continuous Dynamical Systems

Philippe Wenk[*] WENKPH@ETHZ.CH
Learning and Adaptive Systems Group
ETH Zürich and Max Planck ETH Center for Learning Systems
Gabriele Abbati[*] GABB@ROBOTS.OX.AC.UK
Department of Engineering Science
University of Oxford
Stefan Bauer STEFAN.BAUER@TUEBINGEN.MPG.DE
Empirical Inference Group
Max Planck Institute for Intelligent Systems, Tübingen
Michael A Osborne MOSB@ROBOTS.OX.AC.UK
Department of Engineering Science
University of Oxford
Andreas Krause KRAUSEA@ETHZ.CH
Learning and Adaptive Systems Group
ETH Zürich
Bernhard Schölkopf BERNHARD.SCHOELKOPF@TUEBINGEN.MPG.DE
Empirical Inference Group
Max Planck Institute for Intelligent Systems, Tübingen

Abstract

Parameter inference in ordinary differential equations is an important problem in many applied sciences and in engineering, especially in a data-scarce setting. In this work, we introduce a novel generative modeling approach based on constrained Gaussian processes and use it to create a computationally and data efficient algorithm for state and parameter inference. In an extensive set of experiments, our approach outperforms its competitors both in terms of accuracy and computational cost for parameter inference. It also shows promising results for the much more challenging problem of model selection.

1. Introduction

Ordinary differential equations (ODEs) are a ubiquitous tool for modeling problems in many quantitative sciences and engineering. While it is often possible to derive the parametric form of the equations using first principles and expert knowledge, most of these systems have no closed-form solution. Thus, one needs to rely on numerical schemes, which can be challenging given that the exact trajectory of the system is usually unknown. Typically, the true trajectory is measured at some discrete time points and is subjected to observational noise.

[*]. The first two authors contributed equally.

Classical numerical approaches iteratively propose new sets of parameters and then evaluate the latter by numerically integrating them and thus obtaining a trajectory that can be compared against the observed data. As argued e.g. by Varah (1982), this procedure can be turned on its head to improve computational performance. In principle, finding good parameters is equivalent to denoising the states, since good ODE parameters will lead to a trajectory that is close to the true one. In particular Varah (1982) first fit a spline curve to the observations, in order to get an estimate of the true trajectory, and subsequently match the state and derivative estimates of said splines to obtain the ODE parameters. This fundamental idea gave rise to a whole family of *gradient matching* algorithms including spline regression, kernel regression and, in a Bayesian setting, Gaussian process regression.

Gaussian processes provide a very natural and theoretically appealing way for smoothing time series, especially as they are very closely related to Kalman filtering (Hartikainen and Särkkä, 2010). However, it is not straightforward to incorporate them in the gradient matching framework. Since the pioneering theoretical work of Calderhead et al. (2009), there has been quite some controversy on how to deploy them effectively. Calderhead et al. (2009) propose a probabilistic modeling scheme and then perform inference using MCMC. Dondelinger et al. (2013) change this probabilistic setup, to achieve a more efficient MCMC sampling procedure, while Gorbach et al. (2017) introduce a computationally efficient inference scheme based on variational inference. Crucially, all these ideas rely on a product of experts heuristic. An alternative approach was formulated by Barber and Wang (2014), but eventually discarded by Macdonald et al. (2015). However, such product of experts heuristics leads to its own theoretical problems. This is shown by Wenk et al. (2018), who propose a new graphical model that circumvents this problem and presented an efficient MCMC-based inference scheme. A further alternative formulation based on variational inference, including the possibility to impose additional inequality constraints on the derivatives, was provided by Lorenzi and Filippone (2018).

Similarly to Gaussian process-based gradient matching, González et al. (2014) and Niu et al. (2016) use kernel ridge regression in a frequentist setting. Aiming directly for point estimates of the parameters, their approaches are naturally faster than alternatives that build on the use of MCMC and Gaussian processes. Nevertheless, they rely on several trade-off parameters to be tuned via cross-validation, which turns out to be practically challenging in a low-data setting.

In our work, we extend and blend these two paradigms to obtain a computationally efficient algorithm, which is able to learn both states and parameters in a low-data setting. In particular, we

- present a novel generative model, phrasing the parameter inference problem as constrained Gaussian process regression,
- provide a data-efficient algorithm that concurrently estimates states and parameters,
- show how all hyperparameters can be learned from data and how they can be used as an indicator for model mismatch and
- provide an efficient Python implementation for public use.[1]

[1] Code available at https://github.com/gabb7/ODIN

2. Background

2.1 Problem Setting

In this paper, we consider K-dimensional dynamical systems whose evolution is described by a set of differential equations parameterized by a vector $\boldsymbol{\theta}$, i.e.

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \boldsymbol{\theta}). \tag{1}$$

The system is observed under additive, zero-mean Gaussian noise at N discrete time-points $\mathbf{t} = [t_0, \dots, t_N]$. We assume that the standard deviation of the noise is constant over time but may be different for each state dimension. The noise is further assumed to be uncorrelated across dimensions and time points. This leads to the following observation model:

$$y_k(t_i) = x_k(t_i) + \epsilon_k(t_i), \qquad \epsilon_k(t_i) \sim \mathcal{N}(0, \sigma_k). \tag{2}$$

2.2 Temporal Regression

In discrete-time settings, it is common to use a difference equation approach, i.e. the dynamical system is modeled via the equation

$$\mathbf{x}(t_{i+1}) = \mathbf{g}(\mathbf{x}(t_i)) \tag{3}$$

where $\mathbf{g}(\mathbf{x})$ is an unknown, time-independent function which can be modeled as a Gaussian process.

In Gaussian process-based gradient matching, we use a completely different approach. Instead of modeling the artificially-designed function $\mathbf{g}$, we directly interpolate the states $\mathbf{x}$ using a Gaussian process, meaning that we model the function $\mathbf{x}$ that maps a time point t_i to the corresponding state vector $\mathbf{x}(t_i)$. For the sake of readability, we will assume $K = 1$ and use $\mathbf{x}$ to denote the values of $x(t)$ stacked across time points:

$$\mathbf{x} = [x(t_1), \dots, x(t_N)] \tag{4}$$

As demonstrated in the experiments section, the extension to $K > 1$ is straightforward: K independent Gaussian processes can be stacked to model each state dimension independently.

While our algorithm should theoretically work using any nonlinear, differentiable regression technique, we choose to use Gaussian processes. Gaussian processes have superb analytical properties (Rasmussen and Williams, 2006) and have recently shown remarkable empirical results in the context of parameter inference for ODEs (Lorenzi and Filippone, 2018; Wenk et al., 2018). Moreover, Gaussian processes are closely connected to kernel ridge regression thanks to the representer theorem (Schölkopf et al., 2001), allowing to draw close connections between our approach and RKHS-based approaches like RKG2/RKG3 introduced by Niu et al. (2016).

3. ODE-Informed Regression

3.1 GP Regression

As in standard GP regression, we start by choosing a kernel function $k_{\boldsymbol{\phi}}$ which is parameterized by a set of hyperparameters $\boldsymbol{\phi}$. Such function is then used to compute a covariance matrix $\mathbf{C}_{\boldsymbol{\phi}}$, whose elements are given by

$$[\mathbf{C}_{\boldsymbol{\phi}}]_{i,j} = k_{\boldsymbol{\phi}}(t_i, t_j). \tag{5}$$

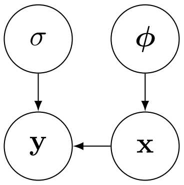


Figure 1: Generative model for standard Gaussian process regression.

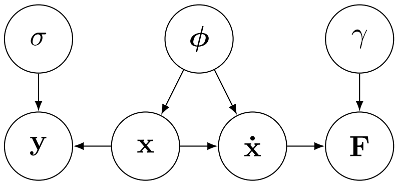


Figure 2: Generative model for Gaussian process regression including derivative observations as used by ODIN.

$\mathbf{C}_{\boldsymbol{\phi}}$ is then used to define a zero-mean prior over the true states $\mathbf{x}$ at the observation times $\mathbf{t}$:

$$p(\mathbf{x} \mid \boldsymbol{\phi}) = \mathcal{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{C}_{\boldsymbol{\phi}}). \tag{6}$$

Using the noise model from equation (2), the likelihood for our observations is a Gaussian given by

$$p(\mathbf{y} \mid \mathbf{x}, \sigma) = \mathcal{N}(\mathbf{y} | \mathbf{x}, \sigma \mathbf{I}). \tag{7}$$

Using Bayes rule and observing the fact that a product of two Gaussians in the same variables is again a Gaussian, we obtain the classic GP posterior

$$p(\mathbf{x} \mid \mathbf{y}, \sigma, \boldsymbol{\phi}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}), \tag{8}$$

where

$$\boldsymbol{\mu}_{\mathbf{x}} = \mathbf{C}_{\boldsymbol{\phi}}(\mathbf{C}_{\boldsymbol{\phi}} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}, \tag{9}$$

$$\boldsymbol{\Sigma}_{\mathbf{x}} = \sigma^2 (\mathbf{C}_{\boldsymbol{\phi}} + \sigma^2 \mathbf{I})^{-1} \mathbf{C}_{\boldsymbol{\phi}}. \tag{10}$$

The graphical representation of this generative model can be found in Figure 1.

3.2 GP Regression with Derivatives

As previously noted e.g. by Solak et al. (2003) and Wenk et al. (2018), the estimate of the posterior distribution of the states given by equation (8) can be further refined if we consider access to noisy observations of the derivatives. Let us then assume we have additional observations $\mathbf{F}$ which are generated by

$$F(t_i) = \dot{x}(t_i) + \delta(t_i), \quad \delta(t_i) \sim \mathcal{N}(0, \gamma). \tag{11}$$

Interestingly, incorporating these derivatives is rather straightforward. Since Gaussian processes are closed under linear operations, the distribution over derivatives is again a Gaussian process. If we condition on the true states, we obtain

$$p(\dot{\mathbf{x}} \mid \mathbf{x}, \boldsymbol{\phi}) = \mathcal{N}(\dot{\mathbf{x}} \mid \mathbf{D}\mathbf{x}, \mathbf{A}), \tag{12}$$

where

$$\mathbf{D} := {}'\mathbf{C}_{\boldsymbol{\phi}} \mathbf{C}_{\boldsymbol{\phi}}^{-1}, \tag{13}$$

$$\mathbf{A} := \mathbf{C}_{\boldsymbol{\phi}}'' - {}'\mathbf{C}_{\boldsymbol{\phi}} \mathbf{C}_{\boldsymbol{\phi}}^{-1} \mathbf{C}_{\boldsymbol{\phi}}' \tag{14}$$

and

$$\left[{}'\mathbf{C}_{\boldsymbol{\phi}}\right]_{i,j} := \frac{\partial}{\partial a} k_{\boldsymbol{\phi}}(a,b)|_{a=t_i,b=t_j}, \tag{15}$$

$$\left[\mathbf{C}'_{\boldsymbol{\phi}}\right]_{i,j} := \frac{\partial}{\partial b} k_{\boldsymbol{\phi}}(a,b)|_{a=t_i,b=t_j}, \tag{16}$$

$$\left[\mathbf{C}''_{\boldsymbol{\phi}}\right]_{i,j} := \frac{\partial^2}{\partial a \partial b} k_{\boldsymbol{\phi}}(a,b)|_{a=t_i,b=t_j}. \tag{17}$$

Equation (12) can now be combined with the likelihood for the derivative observations

$$p(\mathbf{F} \mid \dot{\mathbf{x}}, \gamma) = \mathcal{N}(\mathbf{F} \mid \dot{\mathbf{x}}, \gamma \mathbf{I}). \tag{18}$$

This leads to the generative model shown in Figure 2. Just like in standard Gaussian process regression, all posteriors of interest can be calculated analytically, since all probability densities are Gaussian distributions in $\mathbf{x}$, $\dot{\mathbf{x}}$ or linear transformations thereof.

3.3 Gaussian process-based gradient matching

Given the model in Figure 2, the main challenge is now to include the mathematical expressions of the ODEs in a meaningful way. In traditional GP-based gradient matching, the ODEs are usually introduced as a second generative model for $\mathbf{F}$ or $\dot{\mathbf{x}}$. The latter is then combined with the Gaussian process model depicted in Figure 2 to establish a probabistic link between the observations $\mathbf{y}$ and the parameters $\boldsymbol{\theta}$. However, the Gaussian process model already fully determines the probability densities of $\mathbf{F}$ and $\dot{\mathbf{x}}$. Thus, the two generative models have to be combined using some heuristic like the product of experts (Calderhead et al., 2009; Dondelinger et al., 2013; Gorbach et al., 2017) or an additional Dirac delta function forcing equality (Wenk et al., 2018).

The resulting, unified generative model is then used by the above algorithms to approximate the posterior of $\mathbf{x}$ and $\boldsymbol{\theta}$ through Bayesian inference techniques, e.g. MCMC (Calderhead et al., 2009; Dondelinger et al., 2013; Wenk et al., 2018) or variational mean field (Gorbach et al., 2017). Inference in these algorithms consists in computing mean and standard deviation of an approximate posterior in order to obtain estimates that include uncertainty. As we shall show in the experiment section, this works well for sufficiently tame dynamics and identifiable systems, but struggles to yield meaningful results if the posteriors become multi-modal. Crucially, identifiable systems with unimodal posteriors for the unknown system parameters are a rare exception in practical applications.

3.4 ODIN Oracle

To solve this issue, ODE-informed regression does not include the ODEs via a separate generative model. Instead, we introduce them at inference time in the form of *constraints*.

We start with the joint density of the Gaussian process described in Figure 2, denoted by $p(\mathbf{y}, \mathbf{x}, \dot{\mathbf{x}}, \mathbf{F} \mid \sigma, \gamma, \boldsymbol{\phi})$. As a results of to the Gaussian observation model for $\mathbf{F}$, $\dot{\mathbf{x}}$ can be marginalized out analytically, leading to

$$p(\mathbf{y}, \mathbf{x}, \mathbf{F} \mid \sigma, \gamma, \boldsymbol{\phi}) = \mathcal{N}(\mathbf{y} \mid \mathbf{x}, \sigma^2 \mathbf{I}) \mathcal{N}(\mathbf{x} \mid \mathbf{0}, \mathbf{C}_{\boldsymbol{\phi}}) \mathcal{N}(\mathbf{F} \mid \mathbf{D}\mathbf{x}, \mathbf{A} + \gamma \mathbf{I}). \tag{19}$$

Assuming fixed values for σ, $\boldsymbol{\phi}$ and γ, this equation can be further simplified by taking the logarithm, ignoring all terms that do not explicitly depend on the states $\mathbf{x}$ and parameters $\boldsymbol{\theta}$ and ignoring multiplicative factors to obtain

$$\begin{aligned}\mathcal{R}(\mathbf{x},\mathbf{F},\mathbf{y}) = \mathbf{x}^T\mathbf{C}_{\boldsymbol{\phi}}^{-1}\mathbf{x} \quad (20)\\ + (\mathbf{x}-\mathbf{y})^T\sigma^{-2}(\mathbf{x}-\mathbf{y}) \quad (21)\\ + (\mathbf{F}-\mathbf{D}\mathbf{x})^T(\mathbf{A}+\gamma\mathbf{I})^{-1}(\mathbf{F}-\mathbf{D}\mathbf{x}) \quad (22)\end{aligned}$$

This objective nicely decomposes into three different terms, each one capturing a different but nonetheless important part of the regression. The contribution of the GP prior can be seen in the first summand of equation (20). It encodes our prior smoothness assumptions about our state trajectories and prevents overfitting of the states to the observations. The second term of equation (21) measures the distance between the interpolated states and the observations. Finally, the third summand of equation (22) quantifies the differences between the ODEs output and the derivatives of the Gaussian process. In sharp contrast to classical frequentist methods (Varah, 1982; Niu et al., 2016), all trade-off parameters are naturally provided by the GP framework, once hyperparameters $\boldsymbol{\phi}$ and noise levels σ and γ are fixed.

It is important to note that the functional form of the contributions of the observation model are preserved distinct in the summands of equation (21). Thus, the Gaussianity of the observation noise is by no means a necessary assumption and switching the observation model is as simple as adjusting equation (21). In our derivation, only the Gaussianity of the observation noise on the derivative observations $\mathbf{F}$, governed by the variance γ, is needed: in this way we can analytically marginalize $\dot{\mathbf{x}}$. As we shall see in the experiments section, this noise model is in principle not necessary in the case of perfect ODEs, but provides an effective mechanism for detecting model mismatch.

Given the previous derivations, it is clear that for fixed $\boldsymbol{\phi}$, γ and σ, minimizing $\mathcal{R}(\mathbf{x},\mathbf{F},\mathbf{y})$ is equivalent to maximizing $p(\mathbf{y},\mathbf{x},\mathbf{F} \mid \sigma,\gamma,\boldsymbol{\phi})$ or $p(\mathbf{x},\mathbf{F} \mid \mathbf{y},\sigma,\gamma,\boldsymbol{\phi})$. Nevertheless, so far we have not specified how to obtain the derivative observations $\mathbf{F}$. If we were to ignore the ODEs and had no access to derivative observations, we could just marginalize out $\mathbf{F}$. The objective function yilded this way would be the same as if we deleted the corresponding term from the risk in equation (22) and the result would be identical to standard GP regression. However, if we did have access to a parametric representation of the system dynamics (the ODEs given by equation (1)) and not considered them, we would lose one important insight: the existence of a parameter vector $\boldsymbol{\theta}$ such that $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t),\boldsymbol{\theta})$. Such knowledge can easily be incorporated in the following constrained optimization problem, which is at the core of ODIN:

$$\begin{aligned}&\text{minimize} && \mathcal{R}(\mathbf{x},\mathbf{F},\mathbf{y}) && (23)\\ &\text{with respect to} && \mathbf{x},\mathbf{F} && (24)\\ &\text{subject to} && \exists\theta : \forall i = 1\dots N f(\mathbf{x}_i,\boldsymbol{\theta}) = \mathbf{F}_i && (25)\end{aligned}$$

This optimization problem is equivalent to the unconstrained problem

$$\begin{aligned}&\text{minimize} && \mathcal{R}(\mathbf{x},\boldsymbol{\theta},\mathbf{y}) && (26)\\ &\text{with respect to} && \mathbf{x},\boldsymbol{\theta} && (27)\end{aligned}$$

where

$$\begin{aligned}\mathcal{R}(\mathbf{x},\boldsymbol{\theta},\mathbf{y}) = \mathbf{x}^T\mathbf{C}_{\boldsymbol{\phi}}^{-1}\mathbf{x} \quad (28)\\ + (\mathbf{x}-\mathbf{y})^T\sigma^{-2}(\mathbf{x}-\mathbf{y})\\ + (\mathbf{f}(\mathbf{x},\boldsymbol{\theta})-\mathbf{D}\mathbf{x})^T(\mathbf{A}+\gamma\mathbf{I})^{-1}(\mathbf{f}(\mathbf{x},\boldsymbol{\theta})-\mathbf{D}\mathbf{x})\end{aligned}$$

The resulting algorithm is shown as Algorithm 1. Since we still rely on an oracle to provide us with a good estimate for γ, we call it ODIN-ORACLE.

Algorithm 1 ODIN-ORACLE

```
**Input:**   $\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(K)}, \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}), \gamma_1, \dots, \gamma_K$
*Step 1: GP regression for each state independently*
**for all** $k \in K$ **do**
    Standardize time $\mathbf{t}$ and observations $\mathbf{y}_k$.
    Fit $\boldsymbol{\phi_k}$ and $\sigma_k$using empirical Bayes, i.e. maximize $p(\mathbf{y}^{(k)}|\mathbf{t}, \boldsymbol{\phi}_k, \sigma_k)$.
    Initialize $\mathbf{x}_k$ using $\boldsymbol{\mu}_k$ from equation (9).
**end for**
*Step 2: Include ODE Information*
Initialize $\boldsymbol{\theta}$ randomly.
Apply L-BFGS-B to solve the optimization problem (26) - (28) and obtain $\hat{\mathbf{x}}, \hat{\boldsymbol{\theta}}$.
**Return:** $\hat{\mathbf{x}}, \hat{\boldsymbol{\theta}}$
```

3.5 ODIN: ODE-Informed Regression

From standard GP regression, we have principled methods, given the observations $\mathbf{y}$, to learn the hyperparameters $\boldsymbol{\phi}$ of the GP kernel and the standard deviation σ of the observation noise. However, γ would still need to be hand-tuned. So is there a principled way of learning γ?

Intuitively, γ captures the model mismatch between the true underlying system and the ODEs. If we had access to the true parametric form, we would expect $\gamma = 0$; on the contrary, we would need a large γ to counteract the effect of a fundamentally wrong model. As the correctness of the model is often not known a priori, a principled way of learning γ from data is needed. This can be done by keeping γ as an optimization variable and solve a slightly different optimization problem:

$$\text{minimize} \qquad \mathcal{R}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{y}, \gamma) \tag{29}$$

$$\text{with respect to} \qquad \mathbf{x}, \boldsymbol{\theta}, \gamma \tag{30}$$

where

$$\begin{aligned}\mathcal{R}(\mathbf{x}, \boldsymbol{\theta}, \mathbf{y}, \gamma) = {} & \mathbf{x}^T \mathbf{C}_{\boldsymbol{\phi}}^{-1} \mathbf{x} \\ & + (\mathbf{x} - \mathbf{y})^T \sigma^{-2} (\mathbf{x} - \mathbf{y}) \\ & + (\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) - \mathbf{D}\mathbf{x})^T (\mathbf{A} + \gamma \mathbf{I})^{-1} (\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) - \mathbf{D}\mathbf{x}) \\ & + \log(\det(\mathbf{A} + \gamma \mathbf{I})).\end{aligned} \tag{31}$$

The last term in Equation (31) comes from the fact that now the contribution of the normalization constant of the last Gaussian in equation (19) can no longer be dropped, as it explicitly depends on the optimization variable γ. Similar to the log-determinant in standard GP regression, this term is an automatic Occam's razor keeping the γ small and only allowing for large γ if there is a serious mismatch between the derivatives from the GP model and the output of the ODEs. We will demonstrate in Section 4.3 how this translates to the problem of model selection and model mismatch detection. The resulting algorithm is summarized as Algorithm 2.

Algorithm 2 ODIN

```
Input:  $\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(K)}, \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$
Step 1: GP regression for each state independently
for all $k \in K$ do
    Standardize time $\mathbf{t}$ and observations $\mathbf{y}_k$.
    Fit $\boldsymbol{\phi_k}$ and $\sigma_k$using empirical Bayes, i.e. maximize $p(\mathbf{y}^{(k)}|\mathbf{t}, \boldsymbol{\phi}_k, \sigma_k)$.
    Initialize $\mathrm{x}_k$ using $\boldsymbol{\mu}_k$ from equation (9).
end for
Step 2: Include ODE Information
Initialize $\boldsymbol{\theta}$ randomly.
Initialize $\gamma_1, \dots, \gamma_K = 10^{-2}$
Apply L-BFGS-B to solve the optimization problem (29) - (31) and obtain $\hat{\mathrm{x}}, \hat{\boldsymbol{\theta}}, \hat{\gamma}_1, \dots, \hat{\gamma}_K$.
Return: $\hat{\mathrm{x}}, \hat{\boldsymbol{\theta}}, \hat{\gamma}_1, \dots, \hat{\gamma}_K$
```

4. Experiments

In the following section, we demonstrate the versatility and performance of ODIN for different problem settings on established benchmark systems. As we create our data sets using numerical simulations, we always have access to the ground truth of both the states x^* and the parameters $\boldsymbol{\theta}^*$. Following Wenk et al. (2018), we employ the trajectory RMSE as a metric to compare the quality of parameter estimates:

Definition 1 (Trajectory RMSE) *Let $\hat{\boldsymbol{\theta}}$ be the estimated parameters of an algorithm. Let $\mathbf{t}$ be the vector of the observation times. Define $\tilde{\mathrm{x}}(t)$ as the trajectory one obtains by integrating the ODEs using the estimated parameters, but the true initial value, i.e.*

$$\tilde{x}(0) = \mathrm{x}^*(0) \tag{32}$$

$$\tilde{x}(t) = \int_0^t f(\tilde{x}(s), \hat{\boldsymbol{\theta}}) ds \tag{33}$$

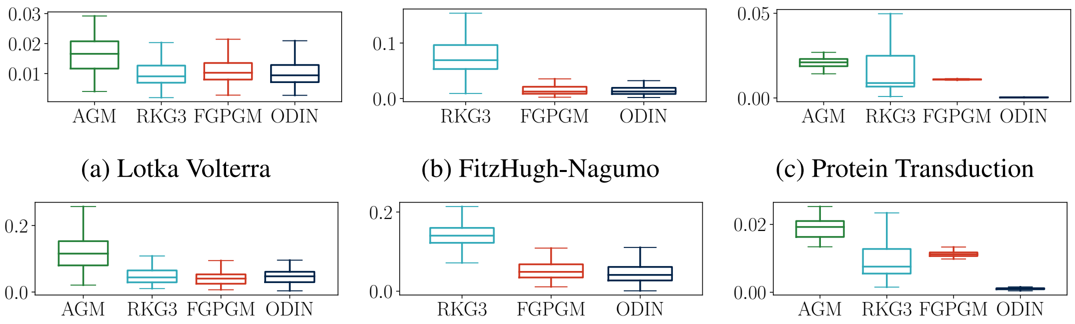
and define $\tilde{\mathrm{x}}$ element-wise as its evaluation at observation times $\mathbf{t}$, i.e. $\tilde{\mathrm{x}}_i = \tilde{x}(t_i)$. The trajectory RMSE is then defined as

$$\textit{tRMSE} := \frac{1}{N} ||\tilde{\mathrm{x}} - \mathrm{x}||_2 \tag{34}$$

where $||.||$ denotes the standard Euclidean norm.

4.1 Benchmark systems

In order to showcase the performance of ODIN, we use four commonly used benchmarking systems. To guarantee a fair comparison, we follow the established parameter settings as used (among others) by Calderhead et al. (2009), Dondelinger et al. (2013), Gorbach et al. (2017) and Wenk et al. (2018). First, we use the Lotka Volterra system originally introduced by Lotka (1932). Due to its locally linear functional form and very smooth dynamics, it is one of the most widely used benchmarks. Second, we choose the FitzHugh Nagumo dynamics originally introduced by FitzHugh (1961) and Nagumo et al. (1962). Its spiky dynamics with fast changing lengthscales provide a formidable challenge for any smoothing based parameter inference scheme. Third, we choose the Protein Transduction



(a) Lotka Volterra (b) FitzHugh-Nagumo (c) Protein Transduction

Figure 3: Trajectory RMSE for three different systems in the parameter inference problem. The top row shows the low noise case with $\sigma = 0.1$ for LV, $SNR = 100$ for FHN and $\sigma = 0.001$ for PT. The bottom row shows the high noise case with $\sigma = 0.5$ for LV, $SNR = 10$ for FHN and $\sigma = 0.01$ for PT.

dynamics originally introduced by Vyshemirsky and Girolami (2007). This system can be seen as the ultimate challenge, with highly nonlinear terms and many previous algorithms claiming only partial parameter identifiability (e.g. Dondelinger et al., 2013; Gorbach et al., 2017; Wenk et al., 2018). Finally, we will rely on the Lorenz 96 dynamics originally introduced by Lorenz and Emanuel (1998) to demonstrate the scaling properties of our algorithm. A more detailed description of all systems can be found in the supplementary material Section A.1.

4.2 State and Parameter Inference

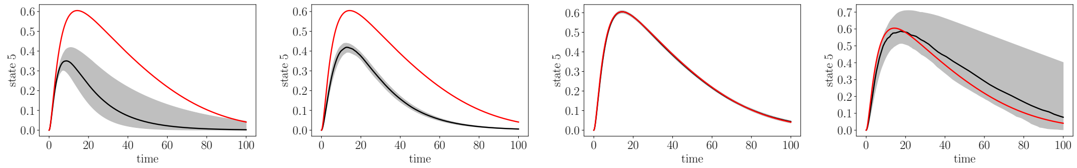
As demonstrated by Solak et al. (2003), including direct observations of $\mathbf{F}$ can drastically improve the accuracy of GP regression. ODIN substitutes these observations by the ODEs, returning both state $\hat{\mathbf{x}}$ and parameter estimates $\hat{\boldsymbol{\theta}}$. It can be shown that including the ODEs improves the quality of the state estimates (cf. Section A.2), but estimating $\hat{\boldsymbol{\theta}}$ is of much greater practical importance. In the following, we will demonstrate that ODIN is capable of learning reliable parameters, outperforming the current state-of-the-art algorithms. We will assume that the true parametric form of our system is provided by a practitioner, derived from first principles and expert knowledge.

4.2.1 Problem Setting

Assume that we are provided with a set of noisy observations $\mathbf{y}$ and the true parametric form $\dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta})$. We then ask ourselves the question of whether it is possible to recover the true parameters $\boldsymbol{\theta}^*$ at observation time.

4.2.2 Parameter Inference

This problem got plenty of attention from the gradient matching community. In this section, we will compare ODIN against AGM (Adaptive Gradient Matching) by Dondelinger et al. (2013), FGPGM (Fast Gaussian Process based Gradient Matching) by Wenk et al. (2018) and RKG3 (RKHS based Gradient matching) by Niu et al. (2016). While both AGM and FGPGM rely on Gaussian process models and MCMC inference, RKG3 employs a kernel regression approach: after modeling the



(a) AGM (b) FGPGM (c) ODIN (d) RKG3

Figure 4: Comparison of the trajectories obtained by numerically integrating the inferred parameters of the Protein Transduction system for the high noise case ($\sigma = 0.01$). The plot was created using 100 independent noise realizations, where the black line is the median trajectory and the shaded areas denote the 25% and the 75% quantiles. The red trajectory shows the ground truth. ODIN is clearly able to handle non-Gaussian posterior marginals yielding state-of-the-art results.

unknown function $x(t)$ as a linear combination of the kernel function at observation times, i.e.

$$\hat{x}(t) = \sum_{i=1}^{N} b_i k(t, t_i) \tag{35}$$

the objective function

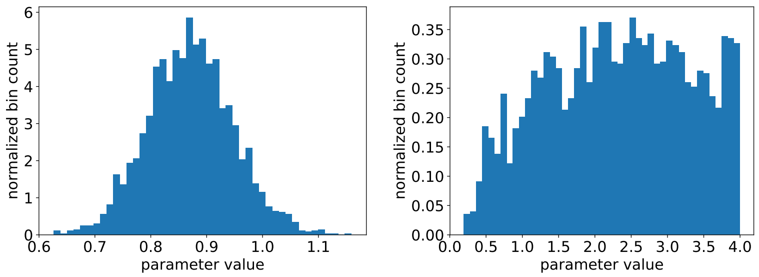
$$\sum_{n=1}^{N} (\hat{x}(t_i) - y(t_i))^2 + \rho \sum_{n=1}^{N} \left(\dot{\hat{x}}(t) - f(\hat{x}(t_i), \boldsymbol{\theta}) \right)^2 \tag{36}$$

is optimized with respect to b_i and $\boldsymbol{\theta}$, where the trade-off parameter ρ has to be determined using cross-validation.

For all comparisons, implementations provided by the respective authors are used. In accordance with current gradient matching literature, the algorithms are evaluated on both a low and a high noise setting for all systems; moreover, the trajectory RMSE, as given in Definition 1, is compared after performing parameter inference. In Figure 3, we show the tRMSE averaged over all states for the three parameter inference benchmark systems. While the average tRMSE is a good indicator for an algorithms overall performance, we also include the statewise tRMSE in section A.4 of the appendix (Figure 15). Unfortunately, AGM was extremely unstable on FitzHugh-Nagumo despite serious hyper-prior tuning efforts on our side. We thus do not show any results for this case. In Figure 4, we compare the trajectories obtained by numerically integrating the inferred parameters. While we can only show a few states in the high noise case of Protein Transduction, a full set of plots can be found in Section A.3.

4.2.3 ODIN VS GP GRADIENT MATCHING

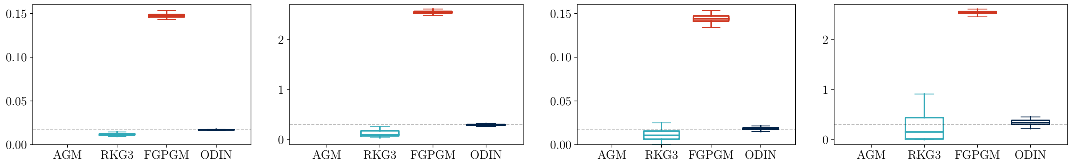
All algorithms perform well on Lotka Volterra, but ODIN clearly outperforms all of its competitors on Protein Transduction. In the Gaussian process setting, this is easily explained by the locally linear form of the Lotka Volterra dynamics (cf. Section 4.1). Both FGPGM and AGM return the mean of the marginal posterior of the parameters at the end of their inference procedure. With locally linear dynamics, we can guarantee that the parameter posterior marginals will be Gaussian distributed. However, no such guarantees can be made in the nonlinear case. As can be seen in Figure 5, the marginal posterior for θ_6 is not Gaussian. In such setting, the mean and the standard deviation of a



(a) Lotka Volterra, θ_4

(b) Protein Transduction, θ_6

Figure 5: Marginals of θ_4 of Lotka Volterra and θ_6 of Protein Transduction for one sample rollout. While the LV marginal is nicely Gaussian, the PT marginal is much wilder.



(a) θ_5, $\sigma = 0.001$

(b) θ_6, $\sigma = 0.001$

(c) θ_5, $\sigma = 0.01$

(d) θ_6, $\sigma = 0.01$

Figure 6: Parameter estimates for Protein Transduction for $\sigma = 0.001$ (left) and $\sigma = 0.01$ (right). Showing median, 50% and 75% quantiles over 100 independent noise realizations.

parameter are less meaningful and it makes much more sense to treat inference as an optimization problem, which is exactly what both ODIN and RKG3 are doing.

4.2.4 IDENTIFIABILITY

Amongst others, both Dondelinger et al. (2013) and Wenk et al. (2018) claim that the last two parameters of Protein Transduction are only weakly identifiable. In fact, this was the main motivation for introducing the trajectory RMSE as a comparison metric, as it directly measures how well a parameter set is able to approximate the true dynamics. Interestingly enough, as shown in Figure 6, neither RKG3 nor ODIN seem to suffer from this problem. This further demonstrates that the expectation of the marginal posterior of the parameters is not the best quantity to infer. While the ratio between θ_5 and θ_6 is fairly stable, the absolute values for AGM have a median magnitude of roughly to 10^{12} and are thus not on the chart.

4.2.5 ODIN VS RKG3

From a conceptual perspective, the main difference between ODIN and RKG3 is the fact that RKG3 forces the states to follow Equation (35), which leads to the derivatives being a deterministic function once the states are fixed. In ODIN, no such assumption is made explicitly. Instead, the states are regularized by a probabilistic Gaussian process prior given by the summand of Equation (20). This Bayesian modeling approach leads to the probabilistic derivative term of Equation (22), where the discrepancies between the GP model and the ODEs are weighted according to the associated uncertainties of the conditional GP prior. Thus, the weighting of the terms in ODIN are a direct

Table 1: Median and standard deviation of computation time (in seconds) for parameter inference over 100 independent noise realizations.

	AGM $[s]$	Niu $[s]$	FGPGM $[s]$	ODIN $[s]$
LV, $\sigma = 0.1$	4548.0 ± 453.8	79.0 ± 19.0	3169.5 ± 90.1	$\mathbf{9.5 \pm 1.3}$
LV, $\sigma = 0.5$	4545.0 ± 558.5	76.5 ± 15.8	3187.5 ± 340.9	$\mathbf{13.2 \pm 6.2}$
FHN, $SNR = 100$	/	74.5 ± 14.3	8678.0 ± 482.7	$\mathbf{2.5 \pm 3.4}$
FHN, $SNR = 10$	/	77.5 ± 12.3	8677.0 ± 487.8	$\mathbf{3.3 \pm 1.3}$
PT, $\sigma = 0.001$	29776.5 ± 4804.7	469.0 ± 21.6	20291.5 ± 435.3	$\mathbf{8.9 \pm 1.5}$
PT, $\sigma = 0.01$	30493.0 ± 1470.4	480.0 ± 42.0	20437.0 ± 713.2	$\mathbf{20.6 \pm 3.75}$

consequence of the underlying probabilistic model. This seems to be a clear advantage over the uniform weighting employed by RKG3 in the second summand of Equation (36), especially since the cross-validation scheme deployed to determine the trade-off parameter empirically seems to be quite fragile.

4.2.6 A NOTE ABOUT PRIORS

While it is quite common to introduce a prior over $\boldsymbol{\theta}$ in a Bayesian inference setting, the graphical model in Figure 2 does not include $\boldsymbol{\theta}$ as a random variable. In a practical setting, we often do not even know the parametric form of the ODEs, and thus it seems quite difficult to justify such a prior. However, it should be noted that our framework can easily accommodate any prior without major modifications. An additional factor $p(\boldsymbol{\theta})$ in Equation (19) directly leads to an additional summand $-\log(p(\boldsymbol{\theta}))$ in Equation (31). From a frequentist perspective, this could be interpreted as an additional regularizer, similar to LASSO or ridge regression. Since all other summands in Equation (31) grow linearly with the amount of observations N and the contribution of the prior stays constant, the regularization term would eventually be dominated in an asymptotic setting.

4.2.7 RUN TIME

In Table 1, we show the training times (in seconds) on the three parameter inference benchmark systems for all algorithms. It is evident (and not unexpected) that the optimization-based algorithms ODIN and RKG3 are orders of magnitude faster than the MCMC based FGPGM and AGM. Furthermore, the need for cross-validation schemes in RKG3 seems to increase its run time roughly by an order of magnitude compared to ODIN.

4.3 Model Selection

If the practitioner is unable to provide the parametric form of our ODEs, he still might be able provide a set of plausible models instead. Of course, these models would then needed to be tested against the observed data.

4.3.1 PROBLEM SETTING

Assume that we are provided with a set of noisy observations $\mathbf{y}$ and a set of candidate models $\dot{\mathbf{x}} = \mathbf{f}_m(\mathbf{x}, \boldsymbol{\theta})$. Is it possible to identify the true parametric form $\mathbf{f}_{m^*}$?

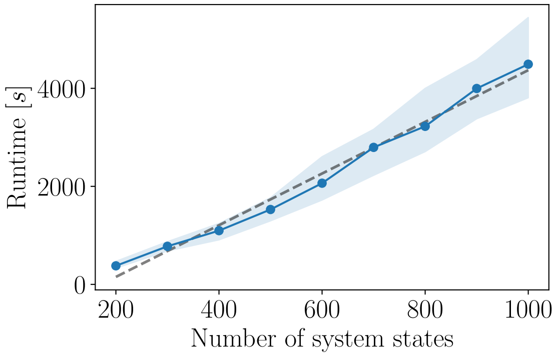


Figure 7: Run time for parameter inference on Lorenz96 for different state dimension including a linear regression fitted to the data. Each system dimension was evaluated using 100 independent noise realizations, plotting both the mean (dots) +- one standard deviation (shaded area).

4.3.2 Results

As stated in Section 3.5, the parameter γ captures the mismatch between the chosen model and the data generating process. As we shall see in this section, this notion can be leveraged for an efficient model selection scheme.

Table 2: Median and standard deviation of γ for different model misspecifications and 100 independent noise realizations.

	$\mathcal{M}_{1,1}$	$\mathcal{M}_{0,1}$	$\mathcal{M}_{1,0}$	$\mathcal{M}_{0,0}$
γ_1	$10^{-6} \pm 0.003$	3.01 ± 0.23	$10^{-6} \pm 0.00$	3.03 ± 0.24
γ_2	$10^{-6} \pm 0.04$	$10^{-6} \pm 0.00$	1.51 ± 0.31	1.53 ± 0.35

For empirical evaluation, we use the Lotka Volterra system described by Equations (39) and (40) to generate our dataset. We then introduce the following model

$$\dot{x}_1(t) = \quad \theta_1 x_1^2(t) + \theta_2 x_2(t) \tag{37}$$

$$\dot{x}_2(t) = \ -\theta x_2(t) \tag{38}$$

and use it to create four different model candidates $\mathcal{M}_{i,j}$, $i, j \in \{0, 1\}$. Here, $i = 0$ indicates that the wrong model is use to model the dynamics of the first state, while $i = 1$ indicates that we used the true parametric form. For example, we construct $\mathcal{M}_{0,1}$ by using equation (37) to model the derivatives of the first state and equation (40) to model the second one. The final γ are presented in Table 2. For numerical reasons, γ was lower bounded to 10^{-6} in all experiments. For a true model, γ settles at this lower bound, while it converges to a much larger value for a wrong model. This also holds dimension-wise for the mixed $\mathcal{M}_{0,1}$ and $\mathcal{M}_{1,0}$, demonstrating that the hyperparameter γ is an efficient tool for identifying true parametric forms.

4.4 Scaling

A key feature of gradient matching algorithms is the linear scaling in the state dimension K. Following Gorbach et al. (2017), we demonstrate this by using the Lorenz96 system with $\theta = 8$, using 50 observations equally spaced over $t = [0, 5]$. The results are shown in Figure 7, including a linear regressor fitted to the means with least squares.

5. Discussion

Parametric ODE models are the backbone of many practical applications. For such models, both Gaussian process and kernel regression-based gradient matching methods have shown to be efficient inference tools. In this paper, we demonstrate how to extend standard GP regression, using theoretical insights to combine the advantages of both algorithm families. The resulting algorithm, namely ODIN, significantly outperforms its competitors in terms of runtime and accuracy for parameter inference while providing an appealing framework for model selection. Unlike its competitors, ODIN has no parameters that need manual tuning, leading to an out-of-the-box applicable tool for both parameter inference and model selection for parametric ODE models.

Acknowledgments

This research was supported by the Max Planck ETH Center for Learning Systems. GA acknowledges funding from Google DeepMind and University of Oxford.

References

David Barber and Yali Wang. Gaussian processes for bayesian estimation in ordinary differential equations. In *International Conference on Machine Learning*, pages 1485–1493, 2014.

Ben Calderhead, Mark Girolami, and Neil D Lawrence. Accelerating bayesian inference over nonlinear differential equations with gaussian processes. In *Advances in neural information processing systems*, pages 217–224, 2009.

Frank Dondelinger, Dirk Husmeier, Simon Rogers, and Maurizio Filippone. Ode parameter inference using adaptive gradient matching with gaussian processes. In *Artificial Intelligence and Statistics*, pages 216–228, 2013.

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

Javier González, Ivan Vujačić, and Ernst Wit. Reproducing kernel hilbert space based estimation of systems of ordinary differential equations. *Pattern Recognition Letters*, 45:26–32, 2014.

Nico S Gorbach, Stefan Bauer, and Joachim M Buhmann. Scalable variational inference for dynamical systems. In *Advances in Neural Information Processing Systems*, pages 4806–4815, 2017.

Jouni Hartikainen and Simo Särkkä. Kalman filtering and smoothing solutions to temporal gaussian process regression models. In *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, pages 379–384. IEEE, 2010.

Edward N Lorenz and Kerry A Emanuel. Optimal sites for supplementary weather observations: Simulation with a small model. *Journal of the Atmospheric Sciences*, 55(3):399–414, 1998.

Marco Lorenzi and Maurizio Filippone. Constraining the dynamics of deep probabilistic models. *arXiv preprint arXiv:1802.05680*, 2018.

Alfred J Lotka. The growth of mixed populations: two species competing for a common food supply. *Journal of the Washington Academy of Sciences*, 22(16/17):461–469, 1932.

Benn Macdonald, Catherine Higham, and Dirk Husmeier. Controversy in mechanistic modelling with gaussian processes. In *International Conference on Machine Learning*, pages 1539–1547, 2015.

Jinichi Nagumo, Suguru Arimoto, and Shuji Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962.

Mu Niu, Simon Rogers, Maurizio Filippone, and Dirk Husmeier. Fast inference in nonlinear dynamical systems using gradient matching. In *Journal of Machine Learning Research: Workshop and Conference Proceedings*, volume 48, pages 1699–1707. Journal of Machine Learning Research, 2016.

CE. Rasmussen and CKI. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA, January 2006.

Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

Ercan Solak, Roderick Murray-Smith, William E Leithead, Douglas J Leith, and Carl E Rasmussen. Derivative observations in gaussian process models of dynamic systems. In *Advances in neural information processing systems*, pages 1057–1064, 2003.

James M Varah. A spline least squares method for numerical parameter estimation in differential equations. *SIAM Journal on Scientific and Statistical Computing*, 3(1):28–46, 1982.

Vladislav Vyshemirsky and Mark A Girolami. Bayesian ranking of biochemical system models. *Bioinformatics*, 24(6):833–839, 2007.

Philippe Wenk, Alkis Gotovos, Stefan Bauer, Nico Gorbach, Andreas Krause, and Joachim M Buhmann. Fast gaussian process based gradient matching for parameter identification in systems of nonlinear odes. *arXiv preprint arXiv:1804.04378*, 2018.

Appendix A.

A.1 Benchmarking Systems

To demonstrate the performance of ODIN, we use four commonly used benchmarking systems. To guarantee a fair comparison, we follow the established parameter settings as used, e.g., by Calderhead et al. (2009), Dondelinger et al. (2013), Gorbach et al. (2017) and Wenk et al. (2018).

A.1.1 LOTKA-VOLTERRA

The Lotka-Volterra system was originally introduced by Lotka (1932) to model population dynamics. It is a 2D system whose dynamics are described by the ODEs

$$\dot{x}_1(t) = \theta_1 x_1(t) - \theta_2 x_1(t) x_2(t) \tag{39}$$

$$\dot{x}_2(t) = -\theta_3 x_2(t) + \theta_4 x_1(t) x_2(t). \tag{40}$$

Using $\boldsymbol{\theta} = [2, 1, 4, 1]$ and initial conditions $\mathbf{x}(0) = [5, 3]$, the system is in a stable limit cycle with a very smooth trajectory. The dataset for this case consists of 20 equally spaced observations on the interval $[0, 2]$. It should be noted that the ODEs are linear in one state or parameter variable if we treat all other state and parameter variables as constants. In the context of Gaussian process based gradient matching, this means that the marginals of the posterior of the parameters and states ar e Gaussian distributed, which makes this system rather easy to handle. The VGM algorithm introduced by Gorbach et al. (2017) is an excellent showcase of how to use this fact to derive an efficient variational approximation.

A.1.2 FITZHUGH-NAGUMO

The FitzHugh-Nagumo model was originally introduced by FitzHugh (1961) and Nagumo et al. (1962) to model the activation of giant squid neurons. It is a 2D system whose dynamics are described by the ODEs

$$\dot{V} = \theta_1 (V - \frac{V^3}{3} + R) \tag{41}$$

$$\dot{R} = \frac{1}{\theta_1} (V - \theta_2 + \theta_3 R). \tag{42}$$

Using $\boldsymbol{\theta} = [0.2, 0.2, 3]$ and initial conditions $\mathbf{x}(0) = [-1, 1]$, this system is in a stable limit cycle. However, the trajectories of this system are quite rough with rapidly changing lengthscales, which is a huge challenge for any smoothing based scheme. Furthermore, both V and θ_1 appear nonlinearly in the ODEs, leading to non-Gaussian posteriors. The dataset for this case consists of 20 equally spaced observations on the interval $[0, 10]$.

A.1.3 PROTEIN TRANSDUCTION

The Protein Transduction model was originally introduced by Vyshemirsky and Girolami (2007) to model chemical reactions in a cell. It is a 5D system whose dynamics are described by the ODEs

$$\begin{aligned}
\dot{S} &= -\theta_1 S - \theta_2 SR + \theta_3 R_S \\
\dot{dS} &= \theta_1 S \\
\dot{R} &= -\theta_2 SR + \theta_3 R_S + \theta_5 \frac{R_{pp}}{\theta_6 + R_{pp}} \\
\dot{R_S} &= \theta_2 SR - \theta_3 R_S - \theta_4 R_S \\
\dot{R_{pp}} &= \theta_4 R_S - \theta_5 \frac{R_{pp}}{\theta_6 + R_{pp}}.
\end{aligned} \tag{43}$$

The parameters and initial conditions of this system were set to $\boldsymbol{\theta} = [0.07, 0.6, 0.05, 0.3, 0.017, 0.3]$ and $\mathbf{x}(0) = [1, 0, 1, 0, 0]$. Due to the dynamics changing rapidly at the beginning of the trajectories, the dataset was created by sampling the system at $\mathbf{t} = [0, 1, 2, 4, 5, 7, 10, 15, 20, 30, 40, 50, 60, 80, 100]$. This system has highly nonlinear terms and many algorithms claim it to be only weakly identifiable (e.g. Dondelinger et al., 2013; Gorbach et al., 2017; Wenk et al., 2018). It can thus be seen as the ultimate challenge for parameter inference algorithms.

A.1.4 LORENZ96

The Lorenz96 system was originally introduced by Lorenz and Emanuel (1998) for weather forecasting. The dimensionality $K > 3$ of this model is completely flexible, with the k-th state being governed by the differential equation

$$\dot{x}_k = (x_{k+1} - x_{k-2})x_{k-1} - x_k + \theta, \tag{44}$$

where all indices should be read modulo K. Introduced to the Gaussian process-based gradient matching community by Gorbach et al. (2017), this system is used to demonstrate the scaling of any algorithm in the amount of states K, by keeping everything but the K fixed. Following Gorbach et al. (2017), we use $\theta = 8$ and 50 equally spaced observations over $t = [0, 5]$.

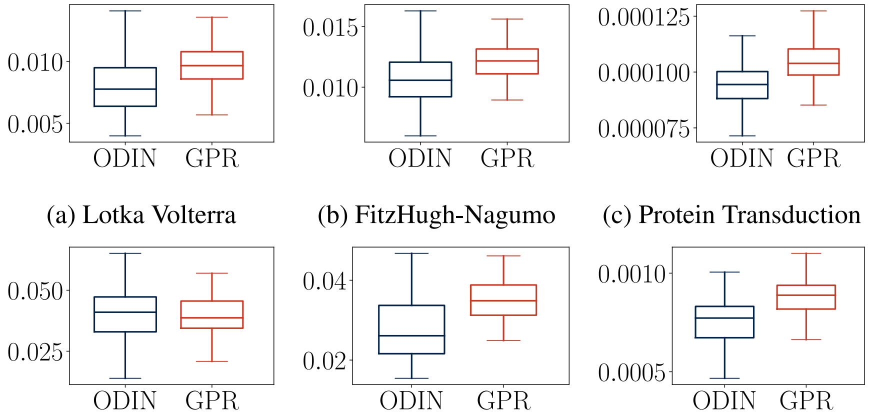
A.2 State Inference

A.2.1 PROBLEM SETTING

Assume that we are provided with a set of noisy observations $\mathbf{y}$ and the true parametric form $\dot{\mathbf{x}} = f(\mathbf{x}, \boldsymbol{\theta})$. Is it possible to recover the true states $\mathbf{x}^*$ at observation times without any information about the true parameters $\boldsymbol{\theta}^*$?

A.2.2 ODES PROVIDE USEFUL INFORMATION

As shown in Figure 8, ODIN provides much more accurate state estimates in all but the high noise LV case compared to standard GP regression. The different behavior can be explained by the quality of the GP prior. For Lotka Volterra, the RBF kernel provides a perfect prior for the sinusoidal form of its dynamics. It is thus not surprising that the GP regression estimates are already quite good, especially in a high noise setting. However, for both FitzHugh Nagumo and Protein Transduction,

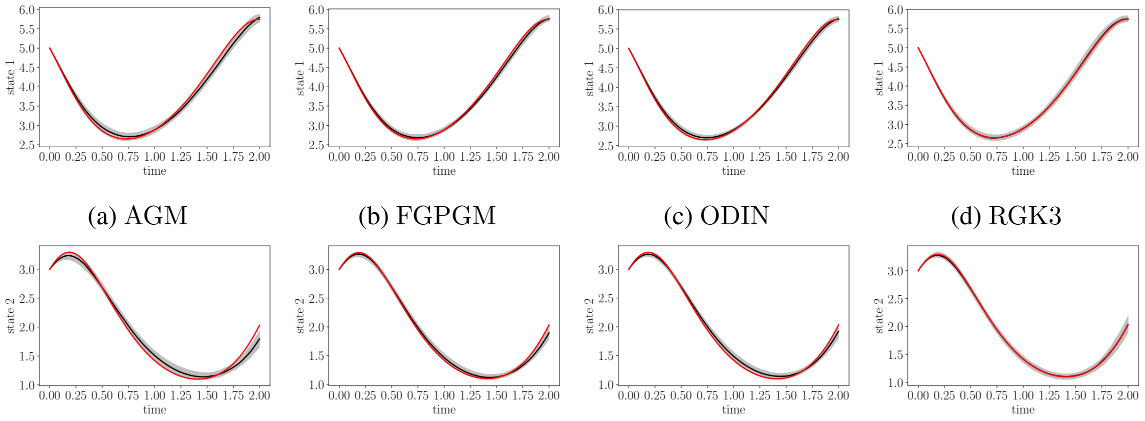


(a) Lotka Volterra (b) FitzHugh-Nagumo (c) Protein Transduction

Figure 8: Comparing the RMSE of state estimates using vanilla GP regression and ODIN. All systems were evaluated on 100 independent noise realizations.

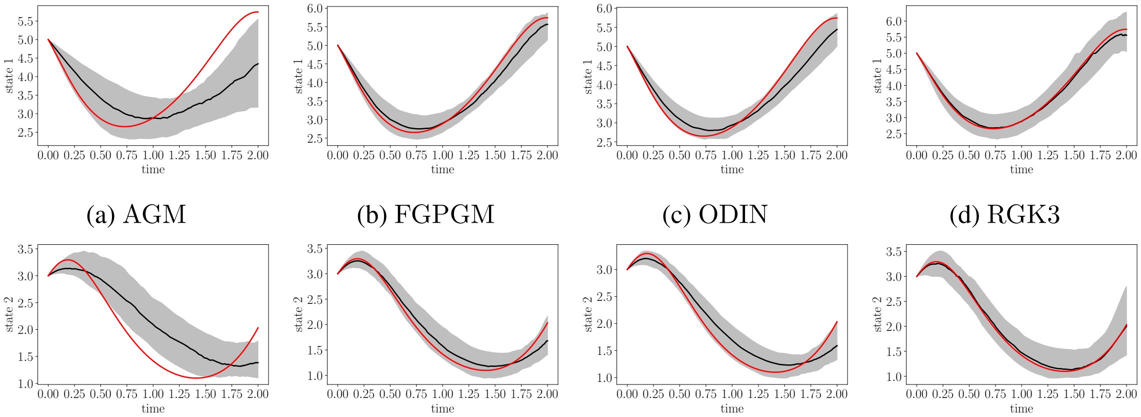
the GP prior is slightly off. Thus, including the additional information provided by the ODEs leads to significant improvements in state estimation.

A.3 Median Trajectories



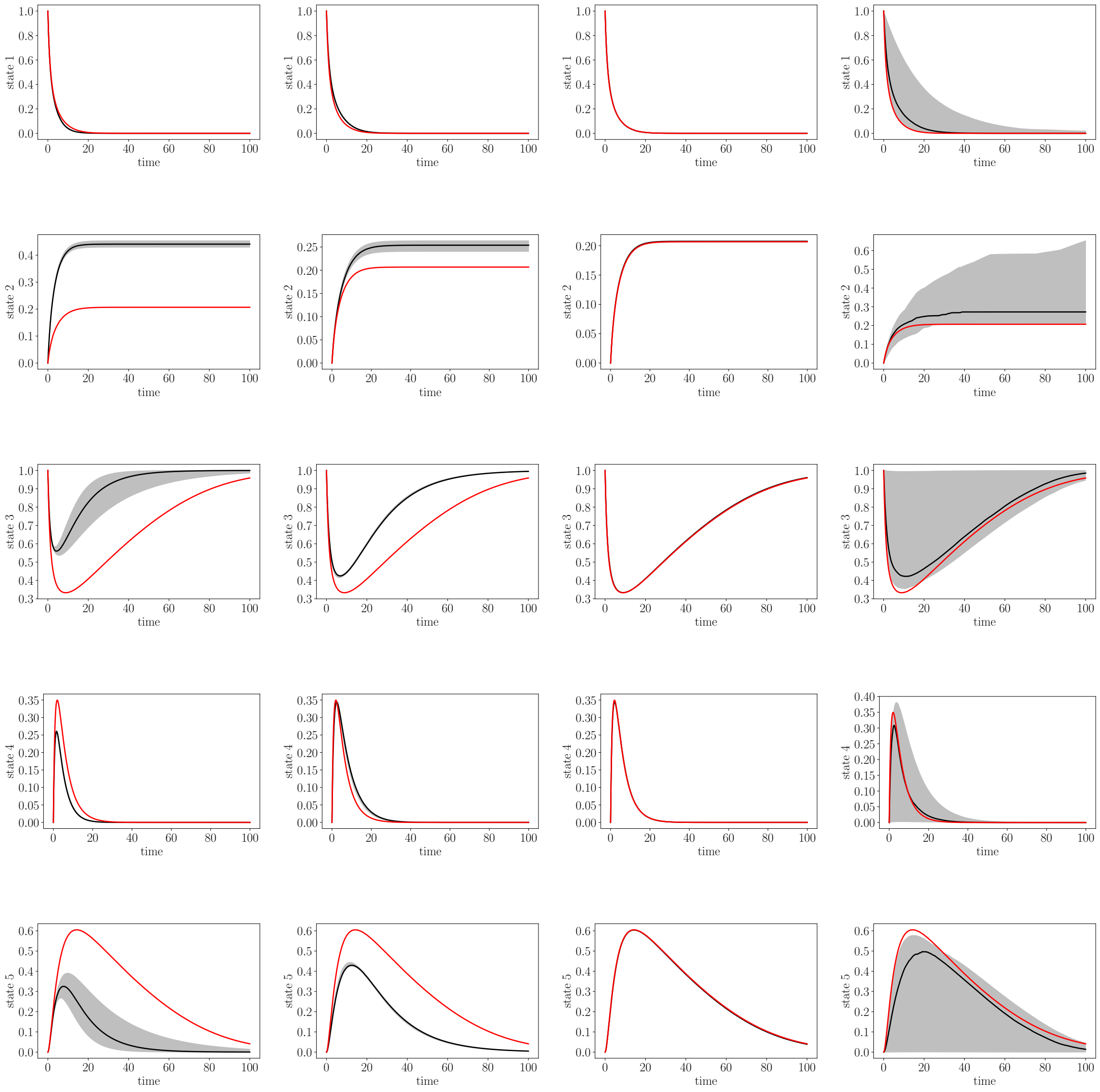
(a) AGM (b) FGPGM (c) ODIN (d) RGK3

Figure 9: Comparing the trajectories obtained by numerically integrating the inferred parameters of the Lotka Volterra system for $\sigma = 0.1$. The plot was created using 100 independent noise realizations, where the black line is the median trajectory and the shaded areas denote 75% quantiles. The red trajectory is the ground truth. The difference between the four algorithms is barely visible.



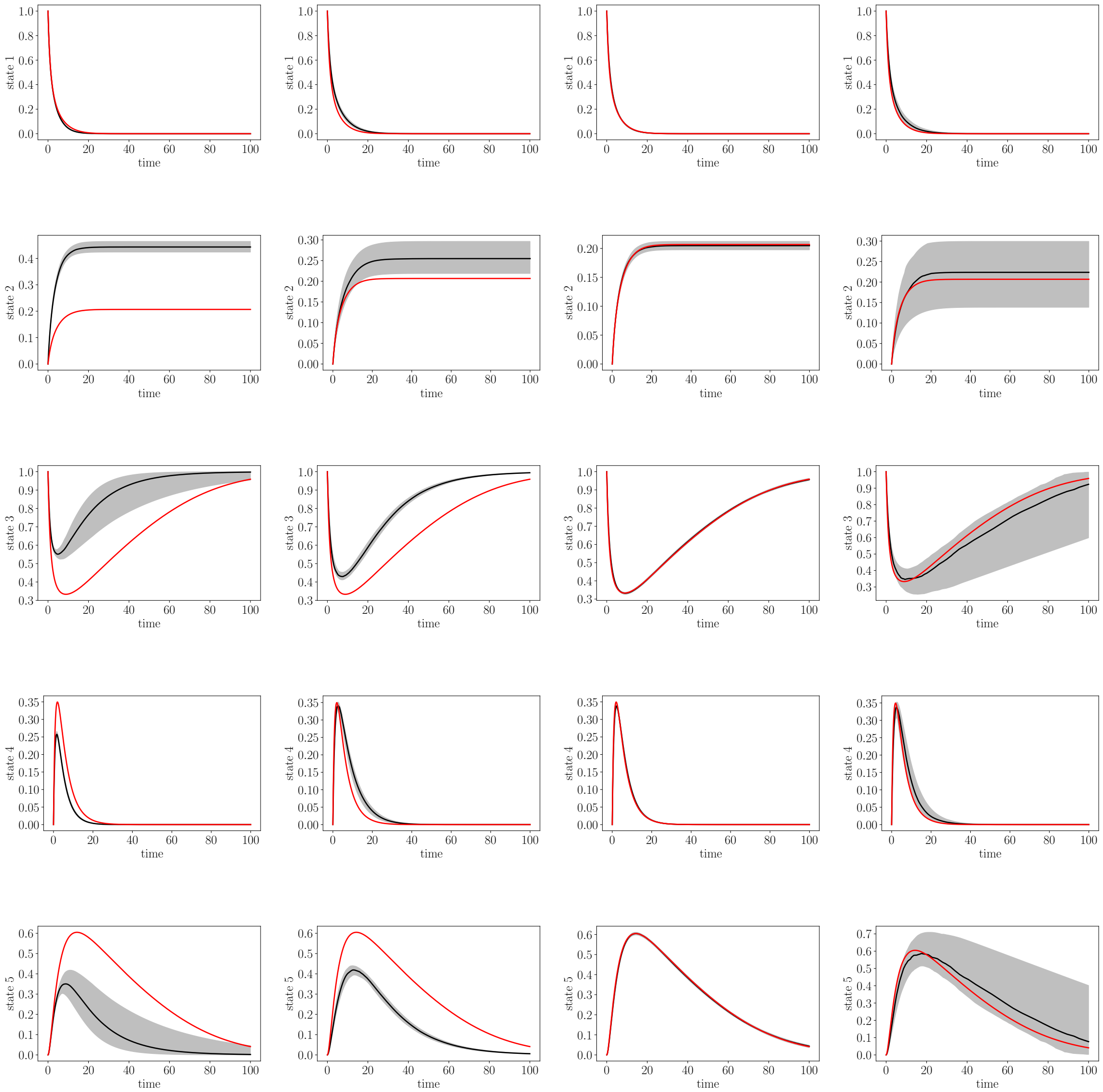
(a) AGM (b) FGPGM (c) ODIN (d) RGK3

Figure 10: Comparing the trajectories obtained by numerically integrating the inferred parameters of the Lotka Volterra system for $\sigma = 0.5$. The plot was created using 100 independent noise realizations, where the black line is the median trajectory and the shaded areas denote 75% quantiles. The red trajectory is the ground truth. While all algorithms seem to perform reasonably well, the perfect match between the sinusoidal dynamics and the RBF kernel lead to a well performing RKG3, while the more flexible Gaussian process based schemes seem to suffer more strongly from a smoothing bias.



(a) AGM (b) FGPGM (c) ODIN (d) RKG3

Figure 11: Comparing the trajectories obtained by numerically integrating the inferred parameters of the Protein Transduction system for $\sigma = 0.001$. The plot was created using 100 independent noise realizations, where the black line is the median trajectory and the shaded areas denote 75% quantiles. The red trajectory is the ground truth. This experiment clearly demonstrates the superior capabilities of ODIN when it comes to handling non Gaussian posterior marginals.



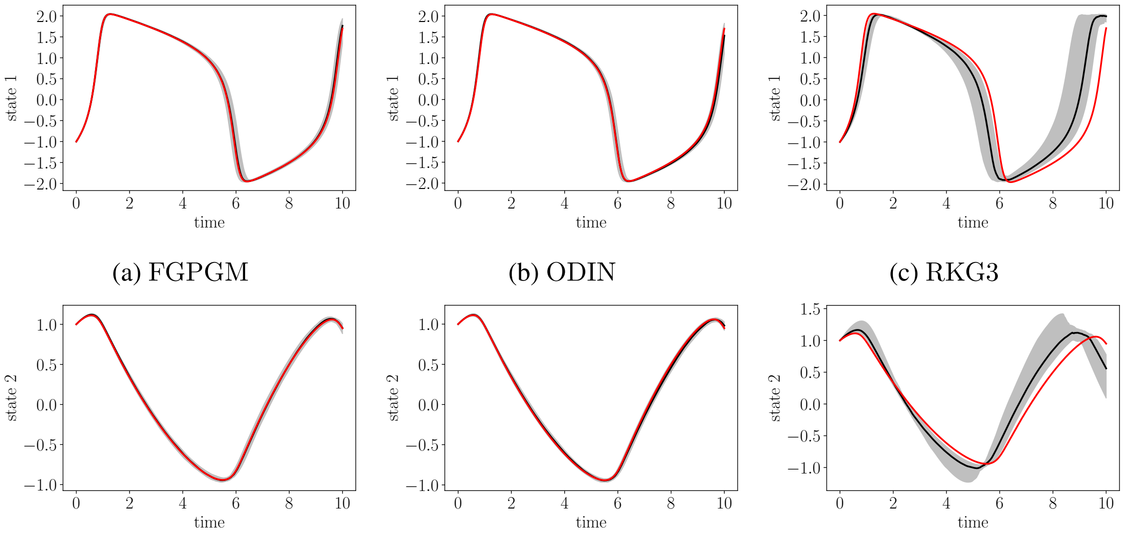
(a) AGM

(b) FGPGM

(c) ODIN

(d) RKG3

Figure 12: Comparing the trajectories obtained by numerically integrating the inferred parameters of the Protein Transduction system for $\sigma = 0.01$. The plot was created using 100 independent noise realizations, where the black line is the median trajectory and the shaded areas denote 75% quantiles. The red trajectory is the ground truth. As in the low noise case, this experiment clearly demonstrates the superior capabilities of ODIN when it comes to handling non Gaussian posterior marginals.

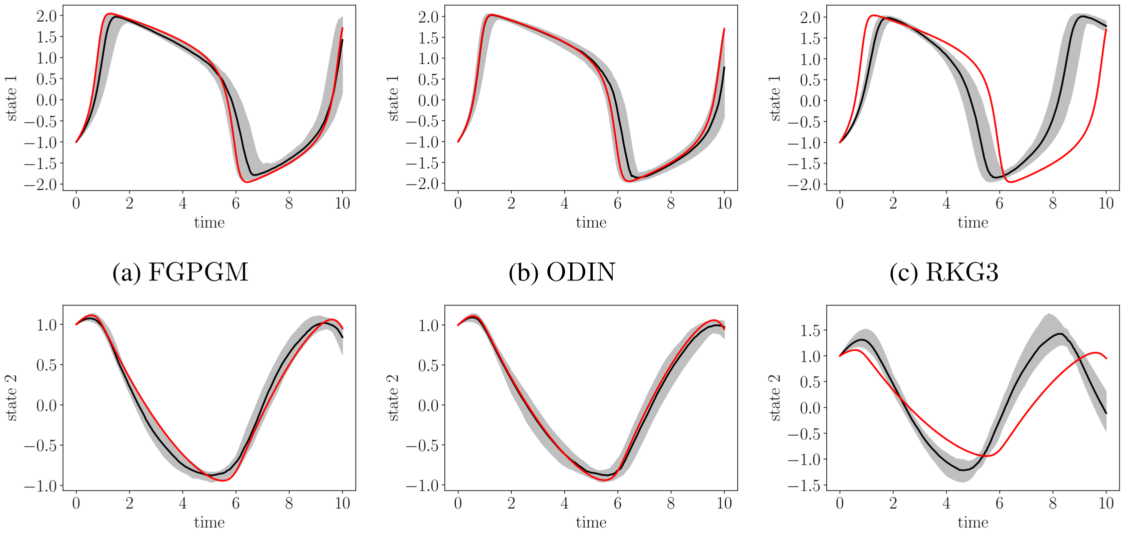


(a) FGPGM

(b) ODIN

(c) RKG3

Figure 13: Comparing the trajectories obtained by numerically integrating the inferred parameters of the FitzHugh-Nagumo system for a SNR of 100. The plot was created using 100 independent noise realizations, where the black line is the median trajectory and the shaded areas denote 75% quantiles. The red trajectory is the ground truth. This experiment clearly shows the superior performance of ODIN when it comes to handling dynamics with rapidly changing lengthscales.



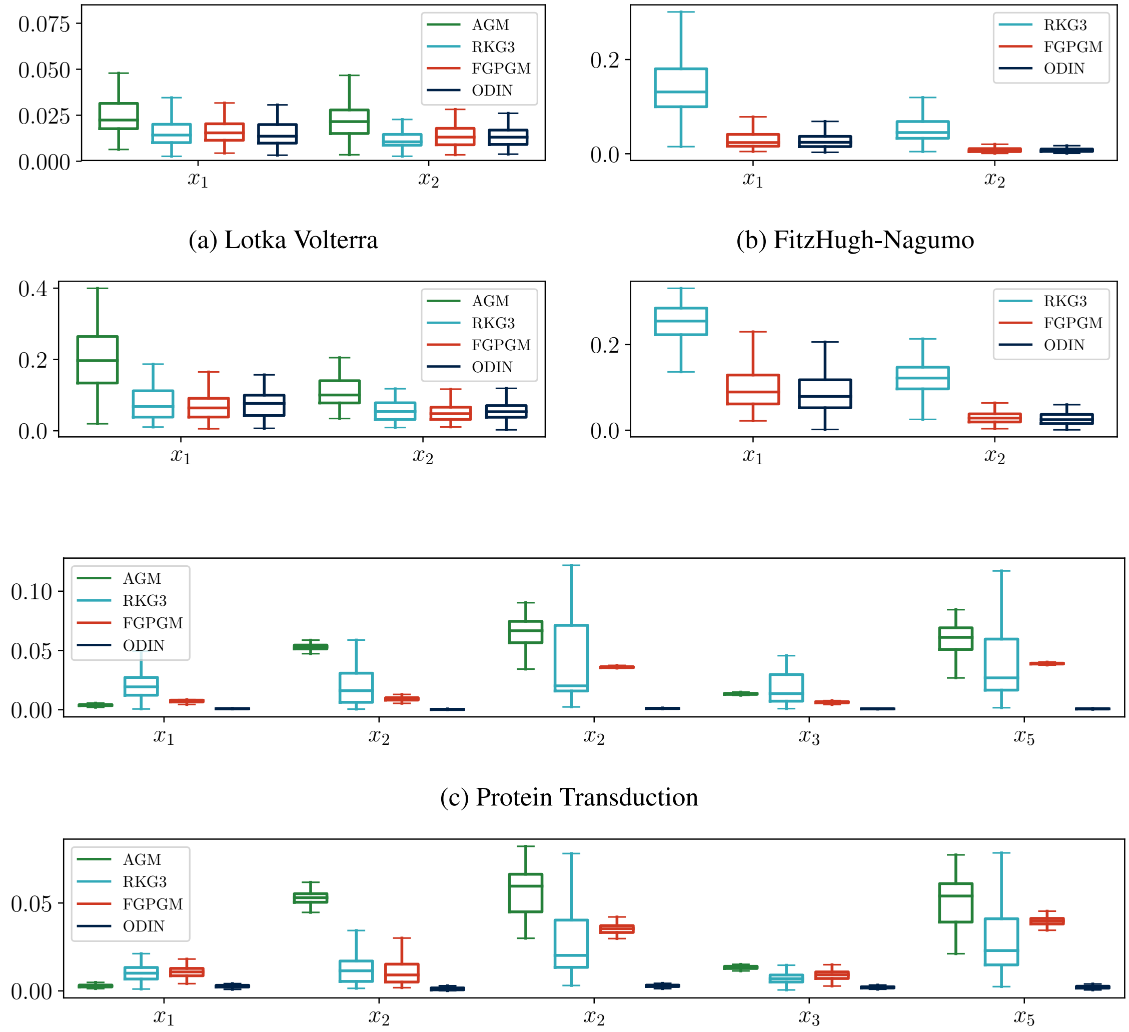
(a) FGPGM

(b) ODIN

(c) RKG3

Figure 14: Comparing the trajectories obtained by numerically integrating the inferred parameters of the FitzHugh-Nagumo system for a SNR of 10. The plot was created using 100 independent noise realizations, where the black line is the median trajectory and the shaded areas denote 75% quantiles. The red trajectory is the ground truth. This experiment clearly shows the superior performance of ODIN when it comes to handling dynamics with rapidly changing lengthscales.

A.4 Statewise tRMSE



(a) Lotka Volterra

(b) FitzHugh-Nagumo

(c) Protein Transduction

Figure 15: Statewise trajectory RMSE for three different systems in the parameter inference problem. For each pair of plots, the top shows the low noise case with $\sigma = 0.1$ for LV, $\sigma = 0.001$ for PT and $SNR = 100$ for FHN. The bottom shows the high noise case with $\sigma = 0.5$ for LV, $\sigma = 0.01$ for PT and $SNR = 10$ for FHN.