White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/202658/

Version: Accepted Version

Proceedings Paper:
Gahungu, P., Lanyon, C.W., Álvarez, M.A. et al. (2022) Adjoint-aided inference of Gaussian process driven differential equations. In: Advances in Neural Information Processing Systems (NeurIPS 2022). 36th Conference on Neural Information Processing Systems (NeurIPS 2022), 28 Nov - 09 Dec 2022, New Orleans, LA, USA. ISBN: 9781713871088. ISSN: 1049-5258.

© 2022 The Authors. This is an author produced version of a paper subsequently published in Advances in Neural Information Processing Systems (NeurIPS 2022). Available under a Creative Commons Attribution 4.0 International License. (https://creativecommons.org/licenses/by/4.0/)

Reuse
This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

arXiv:2202.04589v4 [stat.ML] 5 Dec 2022

Adjoint-aided inference of Gaussian process driven differential equations

Paterne Gahungu
College of Computing
Makerere University

Christopher W. Lanyon
Department of Computer Science
University of Sheffield

Mauricio A. Álvarez
Department of Computer Science
University of Manchester

Engineer Bainomugisha
College of Computing
Makerere University

Michael Smith
Department of Computer Science
University of Sheffield

Richard D. Wilkinson
School of Mathematical Sciences
University of Nottingham

Abstract

Linear systems occur throughout engineering and the sciences, most notably as differential equations. In many cases the forcing function for the system is unknown, and interest lies in using noisy observations of the system to infer the forcing, as well as other unknown parameters. In differential equations, the forcing function is an unknown function of the independent variables (typically time and space), and can be modelled as a Gaussian process (GP). In this paper we show how the adjoint of a linear system can be used to efficiently infer forcing functions modelled as GPs, using a truncated basis expansion of the GP kernel. We show how exact conjugate Bayesian inference for the truncated GP can be achieved, in many cases with substantially lower computation than would be required using MCMC methods. We demonstrate the approach on systems of both ordinary and partial differential equations, and show that the basis expansion approach approximates well the true forcing with a modest number of basis vectors. Finally, we show how to infer point estimates for the non-linear model parameters, such as the kernel length-scales, using Bayesian optimisation.

1 Introduction

Linear systems are used as models throughout the sciences and engineering, encompassing a wide range of both ordinary and partial differential equations (including the heat, wave, Schrödinger's, Maxwell's equations etc), as well as systems of linear algebraic equations (such as eigenvalue problems). To fix notation, let $\mathcal{L} : \mathcal{U} \to \mathcal{V}$ be a linear operator between Banach spaces $\mathcal{U}$ and $\mathcal{V}$ (i.e., complete normed spaces, Kreyszig (1991)) . A prototypical linear system is then of the form

$$\mathcal{L}u = f, \tag{1}$$

where $u \in \mathcal{U}$ is the quantity of interest being modelled, and $f \in \mathcal{V}$ is the *forcing function* of the system. Given a fully specified operator $\mathcal{L}$ and forcing function f (and possibly initial and boundary conditions), solving the system for u is referred to as the *forward problem*. Typically, this is a computationally intensive task. For example, consider modelling air pollution as it moves through the atmosphere. In this case, $\mathcal{L}$ may be a partial differential operator describing the advection, diffusion,

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

and reaction of the pollution, and f will be a function describing the source of the pollution at each location and time. The forward problem refers to computing the concentration, u, given the emission sources and will usually require the use of numerical integration methods.

In many applications, both the linear operator $\mathcal{L}$ and the forcing f may not be fully specified, and we may face the statistical task of learning $\mathcal{L}$ and f from noisy observations of u:

$$z = h(u) + \epsilon. \tag{2}$$

Here, $z \in \mathbb{R}^n$ are the observations, h the observation operator, n the number of observations, and $\epsilon \in \mathbb{R}^n$ a zero-mean observation error. This is often referred to as the *inverse problem* in applied maths and statistics (Stuart, 2010), or sometimes as a *latent force model* in machine learning (Alvarez et al., 2009). In the air pollution example this would equate to finding the distribution of pollution sources, f, given a set of concentration measurements z.

We focus on the situation where

1. f is modelled as a Gaussian process (GP) when $\mathcal{V}$ is an infinite dimensional Banach space, or with a Gaussian distribution in the finite dimensional case. E.g., if $\mathcal{L}$ is an ordinary differential operator with independent variable t, then f will be an unknown function of t.
2. The observation operator $h : \mathcal{U} \to \mathbb{R}^n$ is affine. In the finite dimensional case, this implies $h(u) = Hu + c$ for some constant matrix H and vector c, but in the infinite dimensional case (when $\mathcal{U}$ is a space of functions) includes pointwise evaluation of u, i.e. $u(x, t)$ for some values of x and t, and integral and derivative observations of u, e.g. $\int u \, \mathrm{d}x\mathrm{d}t$, $\frac{\mathrm{d}u}{\mathrm{d}t}$ etc.
3. The observation error ϵ has a Gaussian distribution. This assumption can be relaxed for maximum likelihood (ML) estimation.

The full specification of the statistical model is then

$$\begin{aligned} &\mathcal{L}u = f, && z = h(u) + \epsilon \\ &f \sim GP(m(\cdot), k(\cdot, \cdot)), && \epsilon \sim \mathcal{N}_n(0, \sigma^2 I_n) \end{aligned} \tag{3}$$

where m and k are the prior mean and covariance (kernel) functions of the GP. Note that the linear system in Eqs. (3) may include initial and boundary conditions for differential operators. Our aim is to infer f (and possibly u, $\mathcal{L}$, and k) given z either via

- ML estimation, by solving the constrained optimization problem

$$\min_f \quad (z - h(u))^\top (z - h(u)) + \alpha ||f||^2_{\mathcal{V}} \qquad \text{s.t. } \mathcal{L}u = f$$

- Bayesian inference, by computing the posterior distribution

$$\pi(f, u|z) \propto \pi(z|u)\pi(u|f)\pi(f). \tag{4}$$

The prior distribution for f, $\pi(f)$, and the regularization parameter, α, play a similar role in the two approaches. To solve either problem numerically is likely to require many solves of the forward problem (Eq. 1). For example, for ML we may seek a solution using numerical optimization (e.g. Arellano Jr et al., 2007; Tröltzsch, 2010), whereas with Bayes, we might use an MCMC scheme (e.g. Kopacz et al., 2009; Cotter et al., 2013; Sengupta et al., 2016; Albani et al., 2021) or a variational approach (Chappell et al., 2008). All of these approaches require multiple solves of the forward problem. Here we develop an approach to minimize the computational cost of inference.

1.1 Contribution

In this paper we show that implementing an adjoint of the linear system can result in much faster statistical inference. Instead of using numerical approaches to solve either the ML or Bayesian inference problem, we can do inference for f at the cost of n forward model solves, where n is the number of data points. In many (but not all) cases, this will incur a substantially lower computational cost than competing methods, such as MCMC. More specifically, we show that

1. if f depends linearly on parameters q, we can estimate q or its distribution analytically, i.e. without resorting to numerical integration methods;
2. if we model f as a Gaussian process, then by using a truncated basis expansion we can efficiently infer the posterior distribution for f.

The paper is structured as follows. In the next section we discuss related work before introducing adjoints in Section 3.1. We derive the main results in Section 3.2, and in Section 3.3 we show how linearizing Gaussian processes via a basis expansion reduces inference for GPs in linear systems to simple linear algebra. Finally, in Section 4 we demonstrate the approach on two linear systems: ordinary (ODE) and partial differential equations (PDE).

2 Related work

The problem defined by Eqs. (3) is often referred to as a *latent force model* (Alvarez et al., 2009, 2013). Alvarez et al. (2009) showed how the posterior distribution (4) can be computed by using the integral formulation of $\mathcal{L}u = f$, i.e. $u(x) = \int G(x - v)f(v)dv$, where $G(\cdot)$ is the Green's function associated with the differential operator $\mathcal{L}$. Due to the linearity of the integral transform, placing a GP prior over f leads to a joint GP over f and u. From this joint GP, the posterior distributions $\pi(u \mid f)$ and $\pi(u)$ can be computed in closed form[1]. However, in many situations, particularly for non-trivial differential equation models, the expressions for the covariances are cumbersome and lead to the use of error functions with complex arguments or functions like the Faddeeva function that can be numerically unstable to compute. Guarnizo & Alvarez (2018) proposed representing f using random Fourier features (RFFs) to reduce the number of integrations necessary to be solved analytically. Here, rather than using Green's functions, we instead use adjoints to write the problem as a linear model and then use a reduced-rank Gaussian process formulation, leading to numerically stable and fast approximations to the posterior distribution.

More specifically, estimation of forcing functions in differential equation models has been extensively studied, for example, in the field of modelling atmospheric advection-diffusion (e.g. Yee, 2008; Borysiewicz et al., 2012; Singh & Rani, 2014; Rajaona, 2016; Yeo et al., 2019) with some authors solving the inverse problem using an adjoint approach in combination with MCMC to compute the posterior distribution (e.g. Yee, 2008; Hwang et al., 2019; Luhar et al., 2020; Albani et al., 2021). Of particular relevance is Hwang et al. (2019) who use a point source model of pollution, and use the adjoint to write the backward finite difference approximation, noting that this can be written as a linear model, where the features are conjugate fields associated with each sensor. MCMC sampling is still a limiting factor, restricting the extension of the approach to more complex situations such as time-varying pollution sources.

Other related work includes the stochastic PDE approach of Lindgren et al. (2011), in which the underlying function u is modelled as a Gauss Markov random field (GMRF), which can then be formulated as a stochastic partial differential equation. This allows finite element methods to be used to efficiently compute the posterior distribution for u given z. Similarly, Hartikainen & Särkkä (2010) exploit the link between GMRFs and dynamical systems to convert inference for u to a form in which Kalman filtering methods can be used, which scale linearly with n. Sigrist et al. (2015) focus exclusively on the advection diffusion PDE considered in Section 4.2, and use white noise for the forcing function, f, to create a stochastic PDE model for u. They use a spectral approach, solving the PDE in the Fourier domain, to develop an efficient algorithm for statistical inference. Whilst attractive, it is difficult to generalize this approach from white noise models of f to correlated Gaussian process models. Jidling et al. (2017) show how to infer statistical inference for linearly constrained systems, i.e., where $\mathcal{L}u = 0$. These approaches all model u, whereas our focus is on inferring the forcing function f.

3 Methods

We first recap the definition of adjoints, before deriving our main result illustrating how they can be used to accelerate inference. We then show how using a truncated basis approximation to Gaussian processes allows us to find the GP posterior distribution without resorting to MCMC methods.

3.1 Adjoints

Recall that $\mathcal{L} : \mathcal{U} \to \mathcal{V}$ is a linear mapping between Banach spaces $\mathcal{U}$ and $\mathcal{V}$. Let $\mathcal{U}^*$ and $\mathcal{V}^*$ denote the dual spaces of $\mathcal{U}$ and $\mathcal{V}$ (Kreyszig, 1991). We can construct the adjoint to a continuous linear

[1]For a more detailed explanation of the Green's function method, please see the supplementary material.

operator $\mathcal{L}$ as follows. Let $v^* \in \mathcal{V}^*$ and define $F : \mathcal{U} \to \mathbb{R}$ by $F : u \mapsto v^*(\mathcal{L}u)$. Then F is a bounded linear functional on $\mathcal{U}$, i.e., $F = u^*$ for some $u^* \in \mathcal{U}^*$. Thus for all $v^* \in \mathcal{V}^*$ we've associated a unique $u^* \in \mathcal{U}^*$:

$$\mathcal{L}^* : v^* \mapsto u^* = v^* \circ \mathcal{L}. \tag{5}$$

We call $\mathcal{L}^*$ the *adjoint* of $\mathcal{L}$, and $\mathcal{L}^*$ is itself a bounded linear operator (Estep, 2004). By construction, we have that for all $u \in U$ and $v^* \in \mathcal{V}^*$

$$(\mathcal{L}^* v^*)(u) = v^*(\mathcal{L}u), \tag{6}$$

a result known as the *bilinear identity*. In the case where $\mathcal{U}$ and $\mathcal{V}$ are also real Hilbert spaces with respect to inner products $\langle \cdot, \cdot \rangle_{\mathcal{U}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{V}}$, then we can identify the dual spaces with their underlying space: by the Riesz representation theorem if $v^* \in \mathcal{V}^*$ then there exists $v \in \mathcal{V}$ such that $v^*(\cdot) = \langle \cdot, v \rangle_{\mathcal{V}}$. In this case, the bilinear identity reduces to its more familiar form

$$\langle \mathcal{L}u, v \rangle_{\mathcal{V}} = v^*(\mathcal{L}u) = (\mathcal{L}^* v^*)(u) = \langle u, \mathcal{L}^* v \rangle_{\mathcal{U}} \tag{7}$$

where we now consider $\mathcal{L}^* : \mathcal{V} \to \mathcal{U}$. We only consider real vector spaces here, resulting in a symmetric inner product.

Generally, the adjoint $\mathcal{L}^*$ will be the same type of operator as $\mathcal{L}$ (e.g. if $\mathcal{L}$ is a differential operator then $\mathcal{L}^*$ will be too), so solving an adjoint system of the form $\mathcal{L}^* v = h$ will have similar computational complexity as solving $\mathcal{L}u = f$. See Estep (2004) for an introduction to adjoints, and Section 4 for examples of adjoint systems.

3.2 Benefits of adjoints

How does the development of an adjoint to a linear system help us perform statistical inference for that system? Consider the situation where uncertainty about the unknown forcing function, f in Eq. (1), can be characterized by a linear dependence upon unknown parameters q. That is, we can write

$$f(\cdot) = \sum_{m=1}^{M} q_m \phi_m(\cdot). \tag{8}$$

In the infinite dimensional case where $\mathcal{U}$ and $\mathcal{V}$ are spaces of functions on some set $\mathcal{X}$, the ϕ_m will also be functions on $\mathcal{X}$. In the finite-dimensional case, the ϕ_m will be vectors of fixed length.

In the situation where the observation operator (2) is linear, then we can write the i^{th} observation as $h_i(u) = \langle h_i, u \rangle$ plus noise, for some $h_i \in \mathcal{U}$. Consider the n different adjoint systems

$$\mathcal{L}^* v_i = h_i \text{ for } i = 1, \ldots, n.$$

Then using the bilinear identity (7) we have that

$$h_i(u) = \langle h_i, u \rangle = \langle \mathcal{L}^* v_i, u \rangle = \langle v_i, \mathcal{L}u \rangle = \langle v_i, f \rangle,$$

i.e., the i^{th} observation is the inner product between the unknown forcing function f and the solution of the i^{th} adjoint system. At first, the introduction of the adjoint doesn't appear to have helped. To evaluate the likelihood (or sum of squares) we have gone from needing a single solve of the forward problem, to requiring the solution to n adjoint systems. The benefit arises if we now use the assumption of a linear dependence upon the parameters, and linearity of inner products:

$$h_i(u) = \langle v_i, \sum_{m=1}^{M} q_m \phi_m \rangle = \sum_{m=1}^{M} q_m \langle v_i, \phi_m \rangle.$$

The complete observation vector z can then be written as

$$z = \begin{pmatrix} \langle v_1, \phi_1 \rangle & \ldots & \langle v_1, \phi_M \rangle \\ \vdots & & \vdots \\ \langle v_n, \phi_1 \rangle & \ldots & \langle v_n, \phi_M \rangle \end{pmatrix} \begin{pmatrix} q_1 \\ \\ q_M \end{pmatrix} + \epsilon = \Phi q + \epsilon \tag{9}$$

where $q \in \mathbb{R}^M$ is the parameter vector, and $\Phi \in \mathbb{R}^{n \times M}$ is the matrix of inner products between the n adjoint solutions and M basis vectors.

This can be recognized as a linear model. Thus, standard results can be used to compute the least squares/ML and Bayesian estimators. For ML, the minimum of $S(q) = (z - h(u))^\top (z - h(u))$ subject to $\mathcal{L}u = f$, is obtained at $\hat{q} = (\Phi^\top \Phi)^{-1} \Phi^\top z$ with $\mathbb{V}\text{ar}(\hat{q}) = \sigma^2 (\Phi^\top \Phi)^{-1}$ in the case where the observation errors ϵ_i are uncorrelated and homoscedastic with variance σ^2. Standard results can be used from regularized least squares if we need to regularize q.

In a Bayesian setting, if we assume *a priori* that $q \sim \mathcal{N}_M(\mu_0, \Sigma_0)$, then the posterior for q given z (and other parameters) is

$$q \mid z \sim \mathcal{N}_M(\mu_n, \Sigma_n) \tag{10}$$

where

$$\mu_n = \Sigma_n \left(\frac{1}{\sigma^2} \Phi^\top z + \Sigma_0^{-1} \mu_0 \right), \quad \Sigma_n = \left(\frac{1}{\sigma^2} \Phi^\top \Phi + \Sigma_0^{-1} \right)^{-1}. \tag{11}$$

See, e.g., O'Hagan & Forster (2004). Note that the solutions of the adjoint equation, v_i, are present in the posterior distribution of q in the matrix Φ.

3.3 Inference of Gaussian forcing functions

We now consider the case where the forcing function is given a Gaussian process prior distribution:

$$f(\cdot) \sim GP(m(\cdot), k(\cdot, \cdot)), \tag{12}$$

where $m(\cdot)$ and $k(\cdot, \cdot)$ are the prior mean and covariance functions respectively (Rasmussen & Williams, 2006). Our approach is to use a reduced-rank representation of the GP, as in Eq. (8), derived by truncating an expansion for k of the form

$$k(x, x') = \sum_{m=1}^{\infty} \phi_m(x) \phi_m(x'). \tag{13}$$

There are many possible choices for the basis vectors $\phi_i(\cdot)$, including the Karhunen-Loève (KL) (Deheuvels & Martynov, 2008), Laplacian (Solin & Särkkä, 2020; Coveney et al., 2020; Borovitskiy et al., 2020), and random Fourier feature (RFF) (Rahimi et al., 2007) expansions. The Karhunen Loève basis is formed by finding the spectral expansion of the integral operator defined in Mercer's theorem:

$$T_k f(x) = \int_{\mathcal{X}} k(x, x') f(x') \mathrm{d}x'.$$

In the case where $\dim(x)$ is small (such as the ODE example below), the eigenfunctions of T_k are easy to compute either analytically (Rasmussen & Williams, 2006, Section 4.3.2, p116) or numerically (Greengard & O'Neil, 2021). Using the eigenfunctions of T_k gives the L^2-optimal approximation (Kosambi, 2016), but the eigenfunctions can be difficult to compute even in three dimensions. A simpler approach that extends easily to higher dimensional problems, is to use RFFs (Rahimi et al., 2007). This relies upon Bochner's theorem, which can be used to express stationary kernels $k(x, x') := k(x - x')$ as the Fourier transform of a positive measure p

$$k(x - x') = \int \exp(-iw(x - x')) \mathrm{d}p(w).$$

If we use an isotropic exponentiated quadratic (EQ) kernel k, then the measure p corresponds to a multivariate Gaussian, and Rahimi et al. (2007) give expressions for the Fourier feature basis vectors that can be used to approximate the Gaussian process:

$$k(x - x') = \tau^2 \exp(-\frac{1}{2\lambda^2}(x - x')^\top (x - x')) \quad \Rightarrow \quad \phi_m(x) = \sqrt{\frac{2\tau^2}{M}} \cos(\frac{1}{\lambda} w_m^\top x + b_m) \tag{14}$$

where $w_m \sim N(0, I)$ and $b_m \sim U(0, 2\pi)$. Substituting this ϕ_m into equation 8 allows us to write the forcing function f as a truncated Gaussian process with an EQ kernel (as such the true Gaussian process is never calculated). The extension to anisotropic kernels is straight-forward.

Although the RFF expansion will require more basis vectors (a larger M) than the KL expansion to achieve the same accuracy, the computational complexity of our adjoint approach is dominated by the number of adjoint solves, not the number of features (which only affects the cost of computing the relatively low-cost Eq. 13), and so including more terms has a minor effect on overall computational cost.

3.4 Time complexity of Algorithm 1

Let G be the number of grid elements, n the number of observations, and M the number of features. There are five stages in our algorithm:

1. Solving the n adjoint systems, which requires $\mathcal{O}(Gn)$ operations.
2. Computing each basis vector, ϕ, over the grid for each feature requires $\mathcal{O}(GM)$ operations.
3. Computing the matrix Φ, requires $\mathcal{O}(GMn)$ operations.
4. Finding $\Phi^\top \Phi$ requires $\mathcal{O}(nM^2)$ operations.
5. Finally solving the matrix inverse will require $\mathcal{O}(M^3)$ operations.

This results in an algorithm that scales linearly in the number of data points, n. Empirically, for the problems we've experimented with, we found computing ϕ over the large grid was the most time consuming step. The overall time complexity is $\mathcal{O}(GMn) + \mathcal{O}(nM^2) + \mathcal{O}(M^3)$, but note that the constants associated with each term are important in most problems.

4 Experiments

4.1 Ordinary differential equation (ODE)

Consider the non-homogeneous linear ODE:

$$p_2 \frac{\mathrm{d}^2 u}{\mathrm{d}t^2} + p_1 \frac{\mathrm{d}u}{\mathrm{d}t} + p_0 u = f(t) \tag{15}$$

on the domain $[0, T]$ with initial conditions $u(0) = u'(0) = 0$. The right hand side, $f(t)$, is the unknown forcing function that we wish to estimate, and p_0, p_1, p_2 are parameters in the linear operator. We model f as a zero-mean GP with an EQ kernel (Eq. 14).We assume observations are obtained as noisy averages over short time windows:

$$h_i(u) = \int_{t_i}^{t_i + \Delta t} \frac{1}{\Delta t} u(t) \mathrm{d}t = \langle u, \tilde{h}_i \rangle \tag{16}$$

where $\tilde{h}_i$ is the indicator function $\tilde{h}_i(t) = \mathbb{I}_{[t_i, t_i + \Delta t]}(t)$. This includes direct measurements of $u(t_i)$ if we set $\tilde{h}_i(t) = \delta(t - t_i)$, the Dirac delta function. To generate synthetic data, we simulate a realization f from the GP model, solve Eq. (15) for $u(t)$, and then simulate n observations from Eq. (16) with $T = 1$, $p_2 = 0.5, p_1 = 1, p_0 = 5$, $\Delta t = \frac{T}{n}$, $t_i = \frac{iT}{n}$, adding zero-mean Gaussian noise with standard deviation $\sigma = 0.1$. For simplicity, we solve the ODE with a simple forward Euler approximation, but higher order schemes can and should be used in real applications. We approximate the GP using Eq. (8), using 200 RFFs generated using Eq. (14) with $\lambda = \sqrt{0.6}$ and $\tau^2 = 4$. The linear operator in this case is

$$\mathcal{L}u = \left(p_2 \frac{\mathrm{d}^2}{\mathrm{d}t^2} + p_1 \frac{\mathrm{d}}{\mathrm{d}t} + p_0 \right) u.$$

To derive the adjoint operator we use the bilinear identity (Eq. 7), and integrate by parts twice:

$$\langle \mathcal{L}u, v \rangle = \int_0^T (\mathcal{L}u) v \mathrm{d}t = \int_0^T (p_2 \ddot{u} + p_1 \dot{u} + p_0 u) v \mathrm{d}t = \int_0^T (p_2 \ddot{v} - p_1 \dot{v} + v) u \mathrm{d}t = \langle u, \mathcal{L}^* v \rangle$$

when $v(T) = \dot{v}(T) = 0$. So the adjoint of $\mathcal{L}$ is

$$\mathcal{L}^* v = \left(p_2 \frac{\mathrm{d}^2}{\mathrm{d}t^2} - p_1 \frac{\mathrm{d}}{\mathrm{d}t} + p_0 \right) v.$$

Note that rather than an initial condition, the adjoint has a final condition: to solve the system we have to integrate backwards in time from $t = T$ to $t = 0$. See Algorithm 1.

Fig. 1 shows the effects of the number of training points n and the number of features M on the posterior distribution of the forcing function. As expected, more data results in a more confident

Algorithm 1 Computing the posterior distribution of q

for $i = 1 \dots n$ **do**
  Solve adjoint system $\mathcal{L}^* v_i = \tilde{h}_i$ with appropriate final and boundary conditions.
end for
for $m = 1 \dots M$ **do**
  Sample an RFF basis vector ϕ_m using Eq. (14).
  for $i = 1 \dots n$ **do**
    Compute $[\Phi]_{im} = \langle v_i, \phi_m \rangle$.
  end for
end for
Compute the posterior distribution for q using Eqs. (10) and (11).

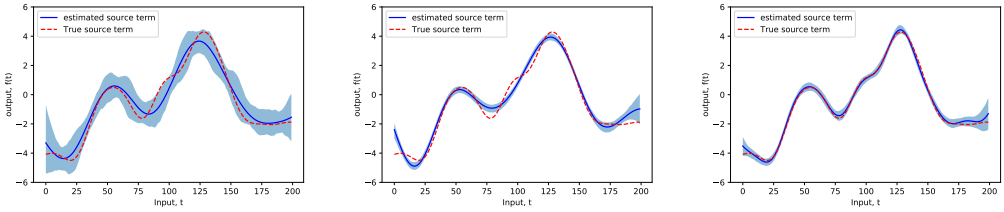


Figure 1: Posterior distribution for the unknown forcing function (with 95% credibility interval). True forcing in red. The number of training points and features are, (left) $n = 10$ and $M = 100$; (middle) $n = 100$ and $M = 10$; (right) $n = 100$ and $M = 100$. The overconfident and wrong posterior when $M = 10$ is a consequence of the model being heavily misspecified.

posterior. Note though the danger of using too few features: with $M = 10$ the approximation of f has limited expressive power and cannot capture the true form of f, i.e., the model is heavily misspecified. This can result in the uncertainty collapsing upon the most likely, but wrong, value. This can be difficult to spot, so users should check the sensitivity of the posterior with respect to M (which can be done with no additional forward solves).

In the supplementary material we provide a comparison between the adjoint method, the Green's function methods as in Alvarez et al. (2009) and Guarnizo & Alvarez (2018), and a vanilla Gaussian process when applied to this ODE problem.

4.2 Partial differential equation (PDE)

We now demonstrate the approach on a PDE in which there are two spatial variables, $x \in \mathcal{X} \subset \mathbb{R}^2$, and time, $t \in [0, T] \subset \mathbb{R}$, so that the solution $u \equiv u(x, t)$ is a function of three independent variables. We consider the advection-diffusion equation

$$\frac{\partial u}{\partial t} + p_1 \cdot \nabla u - \nabla \cdot (p_2 \nabla u) = f \text{ in } \mathcal{X} \times [0, T] \tag{17}$$

with initial and boundary conditions $u(x, 0) = 0$ for $x \in \mathcal{X}$ and $\nabla_n u = 0$ for $x \in \partial\mathcal{X}$. Here, the unknown forcing function $f \equiv f(x, t)$ is a function of space and time, and we model it as a zero-mean Gaussian process, $f(x, t) \sim GP(0, k((x, t), (x', t'))$ with EQ kernel k (Eq. 14). We use an RFF approximation to k with random weights $w_m \sim \mathcal{N}_3(0, I)$ and $b_m \sim U(0, 2\pi)$.

Observations are assumed to arise from *sensors* which take an average of $u(x, t)$ over a small spatial and temporal window $\mathcal{R}_i \times \mathcal{T}_i \subset \mathcal{X} \times [0, T]$:

$$z_i = \langle u, \tilde{h}_i \rangle + e_i \quad \text{with} \quad \tilde{h}_i = \begin{cases} \frac{1}{|\mathcal{R}_i| . |\mathcal{T}_i|} & \text{if } x \in \mathcal{R}_i \text{ and } t \in \mathcal{T}_i \\ 0 & \text{otherwise.} \end{cases}$$

The adjoint of the linear operator $\mathcal{L}u = \frac{\partial u}{\partial t} + p_1 . \nabla u - \nabla \cdot (p_2 \nabla u)$ is

$$\mathcal{L}^* v = -\frac{\partial v}{\partial t} - p_1 . \nabla v - \nabla \cdot (p_2 \nabla v).$$

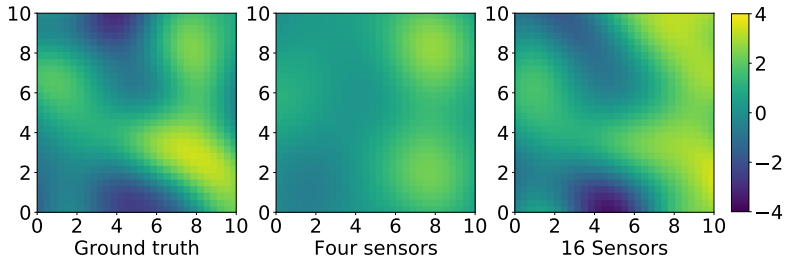


Figure 2: Spatial maps of the forcing function at a single slice: $f(x, 5)$. Left shows ground truth, middle shows the posterior mean with 4 sensors, right shows the posterior mean using 16 sensors.

Table 1: Posterior mean and standard deviation of the q parameters estimated by MCMC and the adjoint method.

	q_1	q_2	q_3	q_4	q_5
MCMC	-3.62 (0.02)	-0.64 (0.01)	1.68 (0.01)	-0.09 (0.01)	4.31 (0.02)
Adjoint	-3.62 (0.02)	-0.64 (0.01)	1.68 (0.01)	-0.09 (0.01)	4.31 (0.02)

Our adjoint-aided approach then requires the solution of

$$\mathcal{L}^* v_i = \tilde{h}_i \text{ in } \mathcal{X} \times [0, T] \text{ for } i = 1, \ldots, n.$$

The final and boundary conditions on the adjoint system are

$$v_i(x, T) = 0 \text{ for } x \in \mathcal{X}, \quad \text{and} \quad p_1 v_i(x, t) + p_2 \nabla v(x, t) = 0 \text{ for } x \in \partial\Omega \text{ and } t \in [0, T].$$

For details of the adjoint derivation see the supplementary material and Estep (2004).

Inference then proceeds as before. After solving the n adjoint systems, we compute the inner product of these solutions with the RFF basis vectors to form the matrix Φ (Eq. 9). We can then use Eq. (11) to compute the posterior with minimal additional computational cost.

Data was simulated on the spatial domain $\mathcal{X} = [0, 10]^2$ for $t \in [0, 10]$ by first randomly generating a forcing function $f(x, t)$ (generated from a GP using an EQ kernel with $\lambda = 2$, $\tau^2 = 2$), and then solving the forward problem (Eq. 17) to find $u(x, t)$ using PDE parameters $p_1 = (0.4, 0.4)$ and $p_2 = 0.01$. We generate n observations using sensors that record averages over short time windows equally spaced across the domain $[0, 10]$ at the locations shown in Fig. 3. Zero-mean Gaussian distributed noise is added to the true sensor readings with standard deviation $\sigma = 0.05$ (note that this is relatively small compared to the signal, which can often create problems for sampling methods). We then use Algorithm 1 to calculate the posterior distribution for q, hence giving the posterior for f. By sampling forcing functions from this posterior and simulating forward, we can evaluate the posterior predictive accuracy of the model.

To validate the posterior estimates from the adjoint method, we also used MCMC to compute the posterior distribution for the PDE model using just $M = 10$ RFFs. Table 1 shows the posterior mean and variance of the first 5 q parameters determined using both methods, which can be seen to be in close agreement. Fig. 7 in the supplementary material shows the trace plots. We used a batch-update random-walk Metropolis-Hastings (MH) sampler, which failed to converge (after an hour of computation) when using a larger number of features. Even with $M = 10$, we still required 10,000s of forward model evaluations to reach convergence (whereas in comparison, the adjoint method required just 75 forward model solves in this case).

To illustrate the effects of sensor density, observations were generated for five time windows using arrays of 4 and 16 sensors at the locations shown in Fig. 3, i.e., a total of either $n = 20$ or $n = 80$ observations. The ground truth forcing and inference results (with $M = 200$) are shown in Fig. 2. As expected, more sensors results in improved estimates. Fig. 3 shows the posterior standard deviation for f in the 4 and 16 sensor cases. Here advection occurs parallel to the $y = x$ line as $p_1 = (0.4, 0.4)$ (i.e., as if there is wind blowing from the south west). We can see that standard deviation is smallest *upwind* of the sensors, with more uncertainty *downwind*, as expected.

To investigate the effects of varying feature and sensor numbers we performed a posterior predictive check using held-out data and used Monte Carlo estimation to calculate the posterior predictive mean

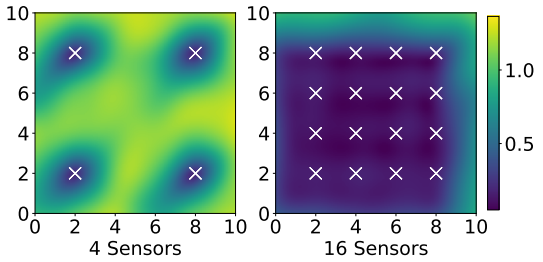


Figure 3: Posterior standard deviation for the examples shown in Fig. 2. The left image is for the case where 4 sensors are used, and the right for 16 sensors. Sensor locations are shown as white crosses.

Table 2: The median MSE as a function of number of sensors and RFFs. The ground truth was generated using a GP with $\lambda = 2$, $\tau^2 = 2$ and PDE parameters $p_1 = (0.01, 0.01)$, $p_2 = 0.01$. The MH algorithm did not converge after 20,000 iterations for 50 or more RFFs. The numbers brackets are the 95% confidence intervals computed from 10 repeated experiments (for the adjoint approach) and 5 (for MH).

Sensors	Features					
	10	50	100	200	300	500
1	3.42 (2.82,4.39)	3.27 (3.13,3.38)	3.24 (3.10,3.37)	3.27 (3.17,3.44)	3.24 (3.12,3.36)	3.27 (3.17,3.35)
4	7.12 (1.57,28.81)	2.39 (2.06,2.62)	2.41 (2.13,2.60)	2.45 (2.32,2.57)	2.50 (2.41,2.69)	2.53 (2.32,2.60)
9	2.38 (1.41,4.40)	2.12 (1.48,3.98)	1.70 (1.49,2.07)	1.48 (1.40,1.72)	1.47 (1.32,1.61)	1.45 (1.40,1.50)
16	1.73 (1.23,3.28)	3.99 (2.32,10.90)	2.18 (1.72,3.54)	1.3 (1.02,1.68)	1.12 (0.98,1.37)	1.12 (1.02,1.21)
25	1.35 (1.19,3.09)	8.93 (4.92,39.86)	4.36 (2.53,8.20)	1.86 (1.43,2.75)	1.35 (1.07,1.81)	1.05 (0.89,1.45)
25 (MH)	3.27 (1.73,6.12)	-	-	-	-	-

squared error (MSE). Table 2 gives the MSE as the number of sensors and RFFs vary. As both increase, so does the accuracy of our estimates. In general, the accuracy depends upon a variety of factors, including the PDE parameters (ratio of diffusion to advection), kernel parameters (decreasing lengthscale makes the problem more challenging), and sensor locations. The speed and efficiency of the proposed adjoint-aided approach allows us to investigate these effects in a way that would not be possible if we were using MCMC (as each estimate of the posterior requires tens of thousands of simulator evaluations, rather than just the n evaluations required for the adjoint-aided approach).

Although not our focus here, we note that we can infer the remaining parameters p (PDE) and (τ^2, λ) (GP hyperparameters) in a variety of ways. For example, using Bayesian optimization we can find the maximum likelihood estimates of these parameters with relatively minimal computational cost. See the supplementary material for details and examples.

In the supplementary material we include some examples of the effect of varying the Gaussian process lengthscale on inference quality, an example in which we apply our method to an advection diffusion problem using the Roundhill II dataset (Cramer & Record, 1957) and apply our method to shift operators (a non-differential linear operator).

4.3 Real time cost analysis

We compared the run time of the adjoint method to a basic MH MCMC algorithm (recorded on a laptop with 16GB RAM and an Intel i7-1065G7 CPU @ 1.50 GHz). Exact inference of the posterior of f in the ODE model using the adjoint method on a 100 element temporal grid with 100 observations using 100 RFFs took 569 ± 72.5 ms. This is approximately the time it took to perform a single iteration of the MH algorithm. Inference of the posterior of f in the PDE model using the adjoint method on a $50 \times 30 \times 30$ spatiotemporal grid with 100 observations using 100 RFFs took 3.32 ± 0.18 s. In this time we were able to run 11 ± 2 iterations of the MH algorithm.

The Metropolis Hastings algorithm is comparatively inefficient due to the matrix multiplication required to compute the forcing function f from a given q using the basis vectors (see equation 8). In the adjoint method this only needs to be calculated once, but is required at each step of the MH algorithm.

Whilst more sophisticated MCMC implementations will achieve faster convergence than random walk samplers, any MCMC algorithm will require 10,000s of forward model evaluations to approximate the same posterior the adjoint method can compute exactly[2] in a fraction of the time.

5 Discussion

This work was motivated by the problem of inferring the distribution of spatially and temporally varying sources of pollution across a city from noisy observations using a model of the pollution's atmospheric transport. Estimating the pollution concentration and its sources can help local authorities reduce the population's exposure, and to motivate policy interventions. Linear systems such as this, are typical of the type of challenge faced throughout the sciences and engineering, and still pose computational challenges that make principled statistical inference almost intractable, or if they are tractable, the computational cost (e.g. of using MCMC) is such that only a limited range of models and situations can be analysed. In this paper we developed an approach that results in conjugate Bayesian inference of an unknown GP forcing function, with a computational cost that scales linearly with the number of observations, n. As n increases, the approach may eventually require more computation than competitor methods such as MCMC (which in theory has cost independent of n), but given that MCMC typically requires $10^5 - 10^6$ iterations even for low-dimensional problems[3], there is still a large range of problems for which an adjoint may be beneficial. In our PDE example, the adjoint-aided approach required orders of magnitude fewer PDE solves than MCMC. We should note that a key limitation of our approach is that it only applies to linear systems and cannot be used to determine forcing functions for non-linear systems such as the Navier-Stokes equations.

We only briefly touched upon the problem of estimating the operator's non-linear parameters p (e.g., advection and diffusion rates) and the GP hyperparameters. *Adjoint sensitivity* methods (see, e.g. Bradley, A. M., 2010; Margossian, 2019; Yashchuk, 2020) can be used to estimate gradients of the log-likelihood with respect to the non-linear parameters in a numerically stable alternative to automatic-differentiation frameworks (such as TensorFlow), which can be unstable when back propagating through long iterative loops. Access to gradient information allows inference of the additional parameters to be performed efficiently within the preferred statistical paradigm (e.g. maximum likelihood using Bayesian optimization - see the supplementary information; Hamiltonian Monte Carlo (Neal, 2011); or a variational autoencoder (Kingma & Welling, 2014) framework etc.).

Finally, there are many ways in which this approach may be accelerated, for example, by the use of intelligent numerical solvers that reuse solution trajectories and adaptive step size solvers, multi-fidelity methods that use varying grid sizes, and stochastic approaches which use only a subset of the data at each stage. The core computational tasks in our approach (solving the adjoint systems) are embarrassingly parallelisable, enabling easy deployment on HPC facilities if required.

Software: The algorithm has been implemented (for both the ODE and PDE problems) in a Python module available at https://github.com/SheffieldML/advectionGP. The repository also contains Jupyter notebooks that are used to produce the figures and tables in this paper.

Acknowledgements

This work was directly funded by EPSRC projects EP/T00343X/1 and EP/T00343X/2. In addition, EB was supported by funding from Google.org and RW by EPSRC projects EP/P010741/1, EP/W000091/1, and EP/X012603/1.

[2]Here "exactly" refers to our ability to write down and compute the posterior for a given basis truncation of the GP, rather than approximating the posterior with MCMC.

[3]In the best case scenario of independent Gaussian posteriors, the number of iterations required for random walk Metropolis Hasting scales as $\mathcal{O}(M^2)$ (Gelman et al., 1997) and $\mathcal{O}(M^{\frac{5}{4}})$ for Hamiltonian Monte Carlo, (Beskos et al., 2013) where M is the dimension of the parameter, with each iteration requiring a solve of the forward problem. For more complex problems, scaling rates will be worse.

References

Albani, R. A., Albani, V. V., Migon, H. S., and Neto, A. J. S. Uncertainty quantification and atmospheric source estimation with a discrepancy-based and a state-dependent adaptative MCMC. *Environmental Pollution*, 290:118039, 2021.

Alvarez, M., Luengo, D., and Lawrence, N. D. Latent force models. In *Artificial Intelligence and Statistics*, pp. 9–16. PMLR, 2009.

Alvarez, M. A., Luengo, D., and Lawrence, N. D. Linear latent force models using Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2693–2705, nov 2013. ISSN 1939-3539. doi: 10.1109/TPAMI.2013.86.

Arellano Jr, A., Raeder, K., Anderson, J., Hess, P., Emmons, L., Edwards, D., Pfister, G., Campos, T., and Sachse, G. Evaluating model performance of an ensemble-based chemical data assimilation system during INTEX-B field mission. *Atmospheric Chemistry and Physics*, 7(21):5695–5710, 2007.

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., and Stuart, A. Optimal tuning of the hybrid monte carlo algorithm. *Bernoulli*, 19(5A):1501–1534, 2013.

Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth, M. P. Matérn Gaussian processes on Riemannian manifolds. *arXiv preprint arXiv:2006.10160*, 2020.

Borysiewicz, M., Wawrzynczak, A., and Kopka, P. Bayesian-based methods for the estimation of the unknown model's parameters in the case of the localization of the atmospheric contamination source. *Foundations of Computing and Decision Sciences*, 37(4):253, 2012.

Bradley, A. M. PDE-constrained optimization and the adjoint method. Online tutorial, `https://cs.stanford.edu/~ambrad/adjoint_tutorial.pd`, 2010. Accessed: 2022-01-27.

Chappell, M. A., Groves, A. R., Whitcher, B., and Woolrich, M. W. Variational Bayesian inference for a nonlinear forward model. *IEEE Transactions on Signal Processing*, 57(1):223–236, 2008.

Cotter, S. L., Roberts, G. O., Stuart, A. M., and White, D. Mcmc methods for functions: modifying old algorithms to make them faster. *Statistical Science*, 28(3):424–446, 2013.

Coveney, S., Corrado, C., Roney, C. H., O'Hare, D., Williams, S. E., O'Neill, M. D., Niederer, S. A., Clayton, R. H., Oakley, J. E., and Wilkinson, R. D. Gaussian process manifold interpolation for probabilistic atrial activation maps and uncertain conduction velocity. *Philosophical Transactions of the Royal Society A*, 378(2173):20190345, 2020.

Cramer, H. E. and Record, F. A. Field studies of atmospheric diffusion and the structure of turbulence. *American Industrial Hygiene Association Quarterly*, 18(2):126–131, 1957.

Deheuvels, P. and Martynov, G. V. A Karhunen–Loeve decomposition of a Gaussian process generated by independent pairs of exponential random variables. *Journal of Functional Analysis*, 255(9): 2363–2394, 2008.

Estep, D. A short course on duality, adjoint operators, Green's functions, and a posteriori error analysis. *Lecture Notes*, 2004.

Gelman, A., Gilks, W. R., and Roberts, G. O. Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, 7(1):110–120, 1997.

Greengard, P. and O'Neil, M. Efficient reduced-rank methods for Gaussian processes with eigenfunction expansions. *arXiv preprint arXiv:2108.05924*, 2021.

Guarnizo, C. and Alvarez, M. A. Fast kernel approximations for latent force models and convolved multiple-output Gaussian processes. In *Uncertainty in Artificial Intelligence*, pp. 835–844, 2018.

Hartikainen, J. and Särkkä, S. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *2010 IEEE international workshop on machine learning for signal processing*, pp. 379–384. IEEE, 2010.

Hwang, Y., Kim, H. J., Chang, W., Yeo, K., and Kim, Y. Bayesian pollution source identification via an inverse physics model. *Computational Statistics & Data Analysis*, 134:76–92, 2019.

Jidling, C., Wahlström, N., Wills, A., and Schön, T. B. Linearly constrained Gaussian processes. *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Kingma, D. P. and Welling, M. Stochastic gradient VB and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, volume 19, pp. 121, 2014.

Kopacz, M., Jacob, D. J., Henze, D. K., Heald, C. L., Streets, D. G., and Zhang, Q. Comparison of adjoint and analytical Bayesian inversion methods for constraining Asian sources of carbon monoxide using satellite (MOPITT) measurements of CO columns. *Journal of Geophysical Research: Atmospheres*, 114(D4), 2009.

Kosambi, D. Statistics in function space. In *DD Kosambi*, pp. 115–123. Springer, 2016.

Kreyszig, E. *Introductory functional analysis with applications*, volume 17. John Wiley & Sons, 1991.

Lindgren, F., Rue, H., and Lindström, J. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

Luhar, A. K., Etheridge, D. M., Loh, Z. M., Noonan, J., Spencer, D., Smith, L., and Ong, C. Quantifying methane emissions from Queensland's coal seam gas producing Surat Basin using inventory data and a regional Bayesian inversion. *Atmospheric Chemistry and Physics*, 20(23): 15487–15511, 2020.

Margossian, C. C. A review of automatic differentiation and its efficient implementation. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 9(4):e1305, 2019.

Neal, R. M. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11): 2, 2011.

O'Hagan, A. and Forster, J. J. *Kendall's advanced theory of statistics, volume 2B: Bayesian inference*, volume 2. Arnold, 2004.

Rahimi, A., Recht, B., et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, pp. 5. Citeseer, 2007.

Rajaona, H. *Inférence Bayésienne adaptative pour la reconstruction de source en dispersion atmosphérique*. PhD thesis, Lille 1, 2016.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

Sengupta, B., Friston, K. J., and Penny, W. D. Gradient-based MCMC samplers for dynamic causal modelling. *NeuroImage*, 125:1107–1118, 2016.

Sigrist, F., Künsch, H. R., and Stahel, W. A. Stochastic partial differential equation based modelling of large space–time data sets. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, pp. 3–33, 2015.

Singh, S. K. and Rani, R. A least-squares inversion technique for identification of a point release: Application to fusion field trials 2007. *Atmospheric environment*, 92:104–117, 2014.

Solin, A. and Särkkä, S. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 30(2):419–446, 2020.

Stuart, A. M. Inverse problems: A bayesian perspective. *Acta Numerica*, 19:451–559, 2010. doi: 10.1017/S0962492910000061.

Tröltzsch, F. *Optimal control of partial differential equations: theory, methods, and applications*, volume 112. American Mathematical Soc., 2010.

Yashchuk, I. Bringing PDEs to JAX with forward and reverse modes automatic differentiation. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.

Yee, E. Theory for reconstruction of an unknown number of contaminant sources using probabilistic inference. *Boundary-layer meteorology*, 127(3):359–394, 2008.

Yeo, K., Hwang, Y., Liu, X., and Kalagnanam, J. Development of hp-inverse model by using generalized polynomial chaos. *Computer Methods in Applied Mechanics and Engineering*, 347: 1–20, 2019.

Supplementary material for *Adjoint-aided inference of Gaussian process driven differential equations*

Latent force models using Green's functions

Existing approaches to latent force models rely upon the concept of a Green's function (e.g. Higdon, 2002; Boyle & Frean, 2005; Alvarez et al., 2009, 2013; Guarnizo & Alvarez, 2018). Here, we briefly describe this approach and how it links to our adjoint-aided approach, and discuss the advantages and disadvantages of both methods. Consider the linear system

$$\begin{aligned} \mathcal{L}u &= f \qquad \text{for } x \in \Omega \\ u &= 0 \qquad \text{for } x \in \partial\Omega. \end{aligned} \tag{18}$$

Here, $\mathcal{L}$ is assumed to be a differential operator, and the solution u is a function of x with domain Ω. The Green's function for this system, $G_y(x)$, satisfies

$$\begin{aligned} \mathcal{L}^* G_y(x) &= \delta_y(x) \qquad \text{for } x \in \Omega \\ G_y(x) &= 0 \qquad \text{for } x \in \partial\Omega \end{aligned} \tag{19}$$

where $\delta_y(x) = \delta(x - y)$ is the Dirac delta function, and $\mathcal{L}^*$ is the adjoint of $\mathcal{L}$. Once we have determined a Green's function, solution of the original problem (18) can be found by computing the convolution of G with f:

$$\begin{aligned} u(y) &= \langle \delta_y, u \rangle && \text{by definition of Dirac delta} \\ &= \langle \mathcal{L}^* G_y,\ u \rangle && \text{by Eq. (19)} \\ &= \langle G_y,\ \mathcal{L}u \rangle && \text{by definition of the adjoint} \\ &= \langle G_y,\ f \rangle && \text{by Eq. (18)} \\ &= \int G_y(x) f(x) \mathrm{d}x. \end{aligned}$$

The standard approach to latent force models then assumes f is a Gaussian process, $f \sim GP(0, k)$, and uses the linearity of this expression and the closure of Gaussian processes under linear operations (Rasmussen & Williams, 2006) to conclude that u is also distributed as a Gaussian process,

$$u \sim GP(0, k_u)$$

with covariance function

$$k_u(y, y') = \int G_y(x) \int G_{y'}(x') k(x, x') \mathrm{d}x' \mathrm{d}x. \tag{20}$$

For some forms of the kernel k, e.g. the exponentiated quadratic kernel, it is possible to evaluate these integrals analytically when G is known. Alternatively, we can resort to numerical integration to evaluate Eq. (20), for example, using random Fourier features (Guarnizo & Alvarez, 2018). Other works have represented G using a polynomial series (Guarnizo & Álvarez, 2018) or have put another GP prior over G (Tobar et al., 2015).

When the Green's function is known for a given system, this approach can work efficiently and may perform as well or better than the adjoint-aided approach. See Cole (2000) for a comprehensive list of Green's functions. However, for many systems (particularly operators with spatially/temporally varying coefficients) the Green's functions are not analytically computable. For diagonalizable operators, we can try to find the eigenfunctions of $\mathcal{L}$, i.e., λ_i, and $\phi_i(x)$ such that $\mathcal{L}\phi_i = \lambda_i \phi_i$, then we can write

$$G_y(x) = \sum_{i=1}^{\infty} \frac{1}{\lambda_i} \phi_i(x) \phi_i(x').$$

If we can estimate λ_i and $\phi_i(x)$ numerically (i.e., by numerically solving the differential equation $\mathcal{L}\phi_i = \lambda_i \phi_i$ on some computational mesh) we can then truncate this sum and form a numerical approximation of $G_y(x)$. But in this case, we would then need to use our numerical approximation of G in a further numerical approximation of the integral in Eq.(20) which can easily lead to numerical

instabilities and low accuracy. In addition, not all differential operators are diagonalizable (i.e., admit a basis of eigenfunctions), for example, operators which are not self-adjoint.

In contrast, our adjoint-aided approach relies solely on the existence of the adjoint operator $\mathcal{L}^*$ and our ability to solve adjoint systems numerically. To do this, we can deploy modern finite element solvers that are efficient, stable, and offer good error-control. A full numerical analysis of the respective errors of the two approaches is beyond the scope of this paper, and would necessarily be specialized to the implementation details of all the particular numerical algorithms used.

In summary, we would recommend that in the special case where G is known and Eq. (20) is tractable, that a Green's function approach be used. In other situations, the ease of the adjoint approach introduced here is likely to be an attractive alternative both in terms of accuracy, numerical stability, and ease of implementation.

Comparison to competing methods

We conducted a comparison between the adjoint method, the Green's function method and a classical Gaussian process on the ordinary differential equation model presented in section 4.1. Observations were taken at 20 time points over $t \in [0, 10]$ with a grid resolution of 200.

- The Gaussian process had a mean squared error of 0.0055 between the true output and the inferred output.
- The Green's function method (as in Alvarez et al. (2009)) had an MSE of 0.0051 for the output error and 0.0860 for the source error.
- The Green's function method with random Fourier features (as in Guarnizo & Alvarez (2018)) achieved the following MSEs:
  - 20 features: Source MSE of 0.099 and output MSE of 0.0058
  - 200 features: Source MSE of 0.0927 and output MSE of 0.0055.
  - 500 features: Source MSE of 0.0856 and output MSE 0f 0.0052.
  - 2000 features: Source MSE of 0.0861 and outpute MSE of 0.0051.
- The adjoint method with M=2000 random Fourier features had an MSE of 0.0056 between the ground truth concentration and the inferred concentration and an MSE of 0.079 between the ground truth and inferred sources.

All three methods achieve a similar quality of inference over the system output. This is to be expected as all three methods utilise a similar statistical model. For larger numbers of features ($M \sim 200$) the adjoint method and the GP predicted the system response with similar accuracy. It should be noted that by using a classical GP approach it is not possible to infer the unknown forcing function, f, which is one of the key advantages of the adjoint method. The Green's function method also performs to a similar level of accuracy as the adjoint method, though the Green's function method with Fourier features appears to perform better at low numbers of features for this particular test case.

Derivation of the Advection Diffusion Adjoint Equation

Consider the advection diffusion operator discussed in Section 4.2:

$$\mathcal{L}u = \frac{\partial u}{\partial t} + \mathbf{p}_1 \cdot \nabla u - \nabla \cdot (p_2 \nabla u) \text{ in } \mathcal{X} \times [0, T] \tag{21}$$

with initial condition

$$u(x, 0) = 0 \text{ for all } x \in \mathcal{X} \tag{22}$$

and Neumann boundary condition

$$\nabla_n u = 0 \text{ for x} \in \partial\mathcal{X}, \tag{23}$$

where $\partial\mathcal{X}$ is the boundary of $\mathcal{X}$, $\nabla_n u = \nabla u \cdot \hat{\mathbf{n}}$ denotes the normal derivative of u, with $\hat{\mathbf{n}}(x)$ the outward facing normal of $\partial\mathcal{X}$ at x. Let $\Omega = \mathcal{X} \times [0, T]$ denote the spatial temporal domain of u.

The adjoint of the system defined by Eqs (21–23) will depend on both the differential operator, and the specific initial and boundary conditions imposed. To derive this, we need to find a linear operator $\mathcal{L}^*$ and a set of boundary conditions so that the bilinear identity

$$\langle \mathcal{L}u, v\rangle = \langle u, \mathcal{L}^* v\rangle$$

is satisfied for all sufficiently smooth functions u and v with compact support in Ω. Let v be such a function, and consider

$$\langle \mathcal{L}u, v\rangle = \int_\Omega \left(\frac{\partial u}{\partial t} + \mathbf{p}_1 \cdot \nabla u - \nabla \cdot (p_2 \nabla u)\right) v \,\mathrm{d}\Omega. \tag{24}$$

In the derivation below, we'll assume $\mathbf{p}_1$ and p_2 are constant, and follow the general steps outlined in Estep (2004). As in the ODE example, the derivation essentially relies upon repeated application of integration by parts. For the first term in Eq. (24):

$$\begin{aligned}
\int_\Omega \frac{\partial u}{\partial t} v \,\mathrm{d}\Omega &= \int_\Omega \frac{\partial}{\partial t}(uv) - u\frac{\partial v}{\partial t}\,\mathrm{d}\Omega \\
&= \int_{\mathcal{X}} \int_0^T \frac{\partial}{\partial t}(uv)\,\mathrm{d}t\,\mathrm{d}x - \int_\Omega u\frac{\partial v}{\partial t}\,\mathrm{d}\Omega \\
&= \int_{\mathcal{X}} u(x,T)v(x,T) - u(x,0)v(x,0)\,\mathrm{d}x - \int_\Omega u\frac{\partial v}{\partial t}\,\mathrm{d}\Omega.
\end{aligned}$$

For the second term in Eq. (24):

$$\begin{aligned}
\mathbf{p}_1 \cdot \int_\Omega v \nabla u &= \mathbf{p}_1 \cdot \left(\int_\Omega \nabla(uv)\,\mathrm{d}\Omega - \int_\Omega u\nabla v\,\mathrm{d}\Omega\right) \\
&= \mathbf{p}_1 \cdot \left(\int_0^T \int_{\mathcal{X}} \nabla(uv)\,\mathrm{d}x\,\mathrm{d}t - \int_\Omega u\nabla v\,\mathrm{d}\Omega\right) \\
&= \mathbf{p}_1 \cdot \left(\int_0^T \oint_{\partial\mathcal{X}} uv\hat{\mathbf{n}}\,\mathrm{d}x\;\,\mathrm{d}t - \int_\Omega u\nabla v\,\mathrm{d}\Omega\right)
\end{aligned}$$

where the first equality uses the vector product rule, and the third the divergence theorem. For the third term in Eq. (24) we have

$$p_2 \int_\Omega v\nabla\cdot\nabla u\,\mathrm{d}\Omega = p_2\left(\int_0^T \oint_{\partial\mathcal{X}} v\nabla u\cdot\hat{\mathbf{n}}\,\mathrm{d}x\,\mathrm{d}t - \int_\Omega \nabla v\cdot\nabla u\,\mathrm{d}\Omega\right).$$

We can then repeat this process on the final term above

$$\int_\Omega \nabla v\cdot\nabla u\,\mathrm{d}\Omega = \int_0^T \oint_{\partial\mathcal{X}} u\nabla v\cdot\hat{\mathbf{n}}\,\mathrm{d}x\,\mathrm{d}t - \int_\Omega u\nabla\cdot\nabla v\,\mathrm{d}\Omega.$$

Combining all of these terms together gives

$$\begin{aligned}
\langle \mathcal{L}u, v\rangle &= \int_\Omega \left(-\frac{\partial v}{\partial t} - \mathbf{p}_1\cdot\nabla v - \nabla\cdot(p_2\nabla v)\right) u\,\mathrm{d}\Omega \\
&\quad + \int_{\mathcal{X}} u(x,T)v(x,T) - u(x,0)v(x,0)\,\mathrm{d}x \\
&\quad + \int_0^T \oint_{\partial\mathcal{X}} uv\mathbf{p}_1.\hat{\mathbf{n}} - p_2 v\nabla u\cdot\hat{\mathbf{n}} + p_2 u\nabla v\cdot\hat{\mathbf{n}}\,\mathrm{d}x\,\mathrm{d}t \\
&= \langle u, \mathcal{L}^* v\rangle + \text{boundary terms}.
\end{aligned}$$

As in the ODE case, we then choose the boundary and initial conditions on v to make the boundary terms above vanish. Firstly, as $u(x,0) = 0$ for all x, setting the final condition $v(x,T) = 0$ for all x eliminates the first boundary term. Secondly, as $\nabla u\cdot\hat{\mathbf{n}}$ is 0 on the boundary (from the boundary conditions on u, Eq. 22), the third term also vanishes. Finally, to set the remainder of the boundary integral to 0 we assume $\mathbf{p}_1 v + p_2\nabla v = 0$ for all $x \in \partial\mathcal{X}$.

Thus our adjoint operator is

$$\mathcal{L}^* v = -\frac{\partial v}{\partial t} - p_1 \nabla \cdot v - \nabla \cdot (p_2 \nabla v) \tag{25}$$

with final condition

$$v(x, T) = 0 \text{ for all } x \in \mathcal{X}$$

and mixed condition

$$\mathbf{p}_1 v + p_2 \nabla v = 0 \text{ for } x \in \partial\mathcal{X}.$$

Note that when solving the original and adjoint systems numerically, checking that the bilinear identity does indeed hold is a useful validation of the derivation and PDE solvers.

PDE Inference Examples

Various factors effect the quality of source inference when using the adjoint method. These include the number of random Fourier features (RFFs), M, the number of observations, n, the locations of the sensors, and the ratio of the ground truth source lengthscale, λ, to the size of the domain. In Section 4.2 we investigated the effect of changing the values of n and M (see Table 2) for a system with a fixed lengthscale. Here we briefly illustrate the effect of changing λ. We consider a $10 \times 10 \times 10$ grid in space and time and two scenarios:

1. 100 sensors arranged in a grid, with readings at 10 points in time, using 1000 RFFs to infer the source ($n = 1000$, $M = 1000$);
2. 16 sensors arranged in a grid, with readings at 5 time points, using 500 RFFs ($n = 80$, $M = 500$).

We generated three ground truth sources using length-scales $l = 5$, 2 and 1. In each case, we used the adjoint method in the scenarios described to infer the posterior distribution of the source. Fig. 4 shows the ground truth generated with $l = 5$ and the inferred source in each scenario at a single time-slice. In this case both models perform similarly on visual inspection. The MSE between the inferred source and the ground truth is 0.004 for scenario 1 ($n = 1000$, $M = 1000$), and 0.008 in scenario 2 ($n = 80$, $M = 500$).

Fig. 5 shows the same information for the source generated with length-scale $l = 2$. In this case, we can visually see that the posterior inference is much more accurate in scenario 1. The MSEs are 0.07 for scenario 1 and 0.68 for scenario 2. Finally, in the case where $l = 1$ (see Fig. 6), visual inspection reveals that in scenario 2, the posterior mean bears little resemblance to the ground truth, whereas key features of the ground truth are visible in the posterior mean for scenario 1. This is reflected in the MSEs, which are 1.85 for scenario 1 ($n = 1000$) and 2.55 for scenario 2 ($n = 80$).

These results demonstrate the expected phenomena, namely that as the ratio of the length-scale to the grid size decreases, more features and observations are required to accurately infer the ground truth. Furthermore, in the short length-scale case the accuracy of inference is generally lower, as the source varies more between sensor locations than in the longer length-scale case. Additionally, in longer length-scale cases, fewer features and observations are required for high quality inference, thus enabling inference with less computational resource.

Finally, Fig. 7 shows the trace plot for the q parameters for an implementation of the Metropolis Hastings algorithm in the case where $M = 10$ features are used. See the main text (Sect. 4.2) for details.

Bayesian Optimisation Output

Although not our focus here, we note that we can infer the remaining parameters $\mathbf{p}_1$, p_2 (PDE) and τ^2, λ (GP hyperparameters) in a variety of ways. By way of illustration, here we use the GPyOpt (González & Dai, 2016) package to maximise the negative log-likelihood using the expected improvement acquisition function in a Bayesian optimization approach (Shahriari et al., 2015).

To perform inference for $\mathbf{p}_1, p_2, \tau^2, \lambda$, we wrote a function which, given a parameter array θ, estimates the posterior predictive accuracy of our source posterior when using θ (here θ may contain a subset

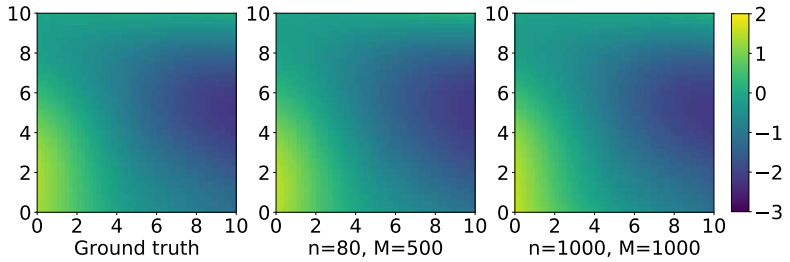


Figure 4: Ground truth and posterior mean of the source when using kernel length-scale, $l = 5$ (time-slice at $t = 5$). The left image shows the ground truth source, the middle image shows the posterior mean inferred using 80 observations and 500 features (MSE=0.008), the right shows the posterior mean inferred using 1000 observations and 1000 features (MSE$= 0.004$).

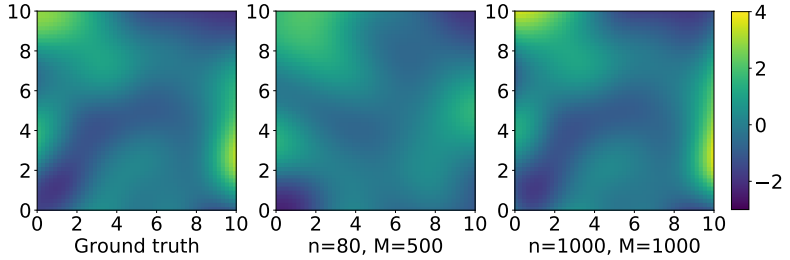


Figure 5: Ground truth and posterior mean of the source when using kernel length-scale, $l = 2$ (time-slice at $t = 5$). The left image shows the ground truth source, the middle image shows the posterior mean inferred using 80 observations and 500 features (MSE=0.68), the right shows the posterior mean inferred using 1000 observations and 1000 features (MSE=0.07).

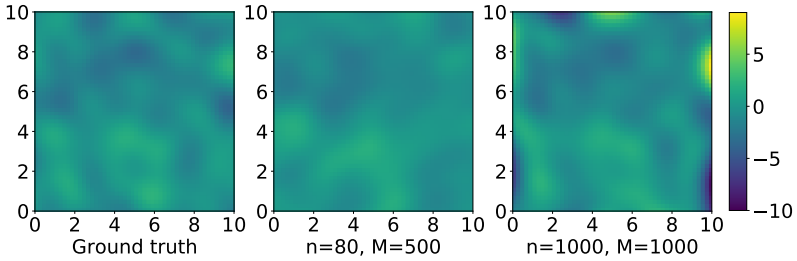


Figure 6: Ground truth and posterior mean of the source from with kernel length-scale, $l = 1$ (time-slice at $t = 5$). The left image shows the ground truth source, the middle image shows the posterior mean inferred using 80 observations and 500 features (MSE=1.85), the right shows the posterior mean inferred using 1000 observations and 1000 features (MSE=2.55).

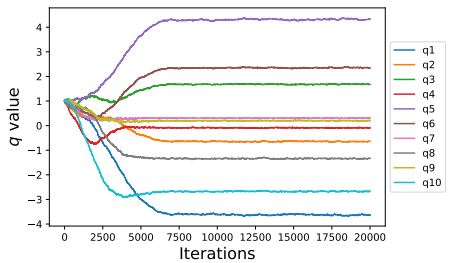


Figure 7: Trace plots for MCMC corresponding to Table 1

of the parameters). We do this by first simulating 100 realizations from the posterior mean of the source for a fixed parameter, i.e., $f_1, \ldots, f_{100} \sim p(f|z, \theta)$, and we push these through the PDE to get a posterior predictive sample of concentration fields $u_1, \ldots, u_{100} \sim p(u|z, \theta)$. The negative log-likelihood is then calculated between these and the training observations giving us a way to score parameter θ. The negative log-likelihood function was used as the objective function in a Bayesian Optimisation routine González & Dai (2016). To test this approach we generated various ground truth source and solution fields using fixed values of θ.

Fig. 8 shows the exploration and eventual convergence in a particular case where we used $\lambda = 2$ to generate a ground truth source. In this case, the maximum likelihood estimate of λ was found to be $\hat{\lambda} = 2.52$ which the optimization found after 29 iterations. In a case where the true kernel length-scale and variance were both 2, and the wind-speed was 0.04, Bayesian optimisation found the maximum likelihood values of 1.24, 3.20 and 0.031 respectively after 20 iterations. Further work is needed to fully explore how to embed this approach into parameter estimation schemes, but we hope it gives some insight into how parameter estimation could be performed. Finally, note that the adjoint approach gives the possibility of estimating the gradient of the loss function. This would enable the adjoint approach to be embedded into gradient based inference algorithms such as the VAE and Hamiltonian Monte Carlo, hopefully accelerating inference.

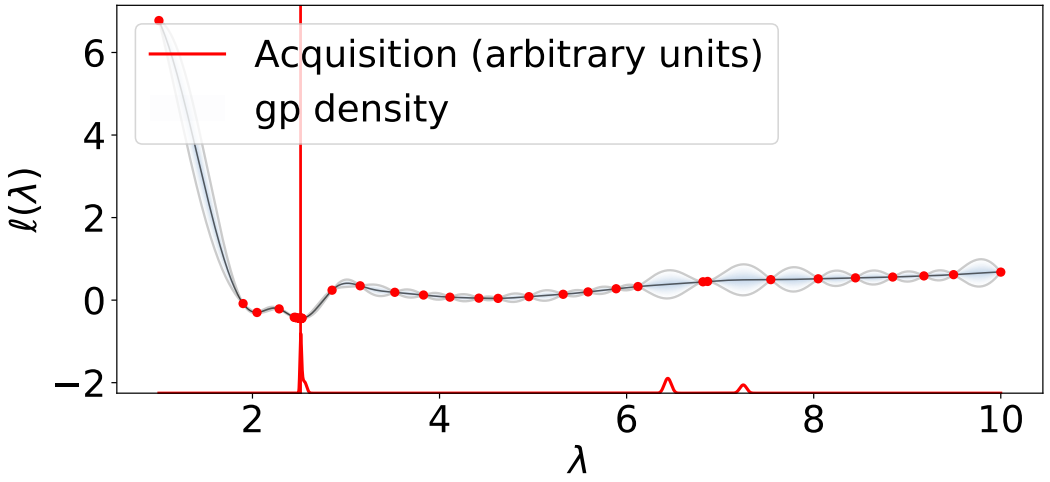


Figure 8: The output of the Bayesian Optimisation algorithm used to infer the value of the GP kernel length-scale, λ. In this case the true value of λ is 2 and the algorithm found the minimum of the negative log-likelihood at $\lambda = 2.52$. This plot shows where the objective function was evaluated, and the posterior mean and variance at each point.

PDE model applied to the Round Hill II dataset

To test the approach using data from a physical experiment, we used the Round Hill II advection-diffusion experiment (Cramer & Record, 1957). In this study, researchers deployed 183 midget impingers for measuring sulphur dioxide in three partial concentric rings, 50m, 100m and 200m downwind from the release site, spanning 69°. A constant source of sulphur dioxide (releasing approximately 5-10 gs^{-1}) was used over a ten minute period, during which the impingers took measurements of average concentration over the first 30 seconds, 3 minutes and 10 minutes. The average wind speed and direction was recorded (2.14 ms^{-1}). We modelled this with our adjoint approach, as in section 4.2, over a $250m \times 250m$ domain spanning 13 minutes, using 10,000 random Fourier bases to approximate the Gaussian process forcing term. We tested two aspects of our model's capabilities. First: Source attribution. The model's mean source prediction was roughly flat except for a peak approximately 45m downwind of the true release site, see Figure 9. This discrepancy is expected as the true dataset contained a point source while our model had a GP prior (with EQ kernel and lengthscale of 10m) over the source. This leads to an inferred broader source, slightly closer to the ring of sensors. The second test was predicting the SO_2 concentration: We removed the middle (100m) ring of sensors from the training data, then tried to predict their measurements.

For comparison we used a Gaussian process with a length-scale of 30m (and 30s) to predict the concentration. We found it useful to threshold the concentrations to be non-negative for both methods. Our model performed considerably better than the GP model. For the three measurement periods the results were:

	Our Model	GP Model
30s	14444	21385
180s	6628	12968
600s	4503	8490

[measurement units were mg/m^3, so these MSEs are in $(mg/m^3)^2$] See figure 10 for a comparison of the inferred concentrations between the two models. Figure 10b indicates that the Gaussian process generally overestimated the right hand side of the left out sensor array, whereas it can be seen from 10a that the adjoint method predicted the true high concentration area fairly well, with some smaller overprediction at the left and right hand sides of the array.

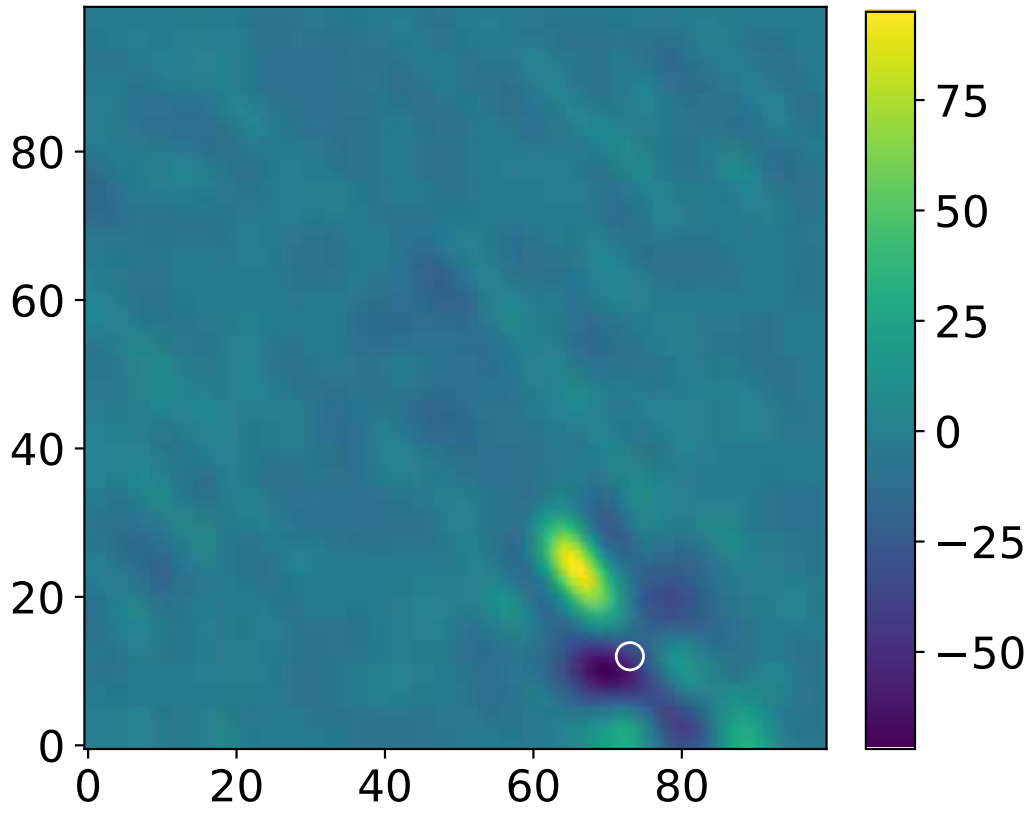


Figure 9: The mean inferred source in the Roundhill experiment at $t = 0$. The white circle indicates the true source location.

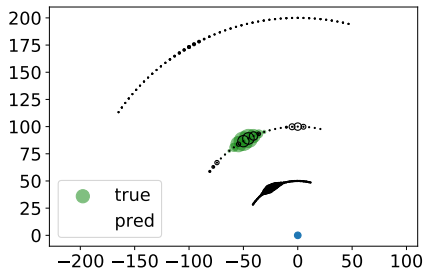
Shift operators

Here we provide an example of the adjoint method applied to a non-differential operator: the shift operator. Consider the operator $\mathcal{L}_a : \mathbb{R} \to \mathbb{R}$ such that

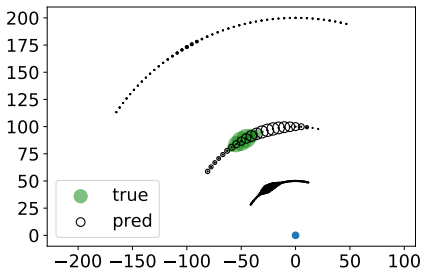
$$\mathcal{L}_a(u(t)) = u(t + a). \tag{26}$$

This is the right shift operator. We can derive the adjoint of the right shift operator by taking the following inner product and using a change of variable (x=t+a).

$$\langle \mathcal{L}u, v \rangle = \int_{-\infty}^{\infty} u(t + a)v(t)dt = \int_{-\infty}^{\infty} u(x)v(x - a)dx \tag{27}$$



(a) The relative pollutant concentrations in the Roundhill experiment between the predicted concentration inferred using the adjoint method and the true values.



(b) The relative pollutant concentrations in the Roundhill experiment between the predicted concentration inferred using a Gaussian process and the true values.

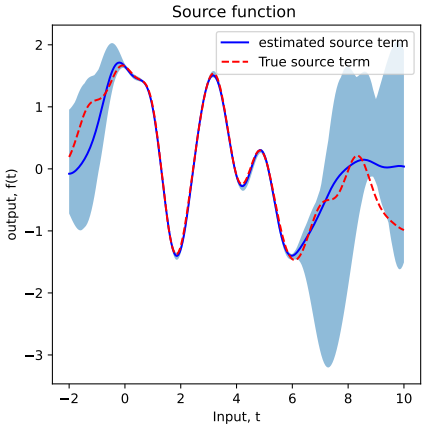
Figure 10

and so the adjoint of the right shift operator is the left shift operator, $\mathcal{L}_a^* : \mathbb{R} \to \mathbb{R}$, where $\mathcal{L}_a^* v(t) = v(t-a)$. Having derived the adjoint of the right shift operator, it is possible to apply the adjoint method to an example system

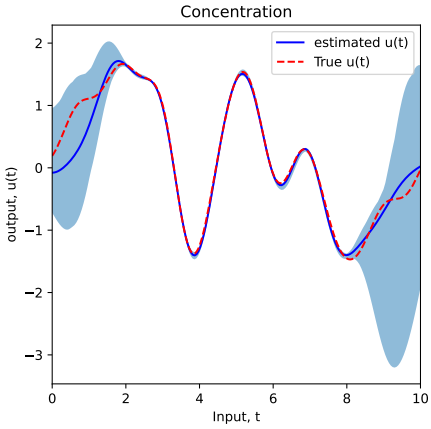
$$\mathcal{L}_a u(t) = u(t+a) = f(t) \tag{28}$$

where $f(t)$ is an unknown forcing function and observations of u are obtained as noisy averages over short time windows (see equation 16).

Figure 11 shows the inferred and true $f(t)$ and $u(t)$ of the system given in equation 28 with $a = 2$ and 20 observations evenly spaced between $t = 2$ and $t = 8$. Observations were generated with Gaussian noise, $\epsilon \sim N(0, 0.05)$. The output $u(t)$ is well predicted within the observation range with relatively high certainty, the prediction is uncertain outside of the observation range. The forcing function, f is well predicted between $t = 0$ and $t = 6$, i.e. the observation range shifted left by $a = 2$. The MSE between the observations and the mean value of $u(t)$ inferred at the observations points is 0.003. For reference, the standard deviation of the observations is 1.001.



(a) Inferred and ground truth source in the shift operator system with $a = 2$



(b) Inferred and ground truth output, $u(t)$ in the shift operator system with $a = 2$

Figure 11

It seems evident that to predict the source and output more confidently over the entire real line would require observations over the entire real line. Furthermore, it does not seem possible to infer the shift parameter, a. For example, if the true shift parameter is a^* and the the model used to infer the source assumes shift parameter a, the inferred source will simply be shifted left by a and the quality of prediction of observation points would be indistinguishable (given a fixed basis for expansion of f).

Supplementary References

Alvarez, M., Luengo, D., and Lawrence, N. D. Latent force models. In *Artificial Intelligence and Statistics*, pp. 9–16. PMLR, 2009.

Alvarez, M. A., Luengo, D., and Lawrence, N. D. Linear latent force models using Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2693–2705, nov 2013. ISSN 1939-3539. doi: 10.1109/TPAMI.2013.86.

Boyle, P. and Frean, M. Dependent Gaussian processes. In Saul, L., Weiss, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2005. URL https://proceedings.neurips.cc/paper/2004/file/59eb5dd36914c29b299c84b7ddaf08ec-Paper.pdf.

Cole, K. Green's Function Library, 2000. URL http://www.engr.unl.edu/~glibrary/glibcontent/glibcontent.html.

Cramer, H. E. and Record, F. A. Field studies of atmospheric diffusion and the structure of turbulence. *American Industrial Hygiene Association Quarterly*, 18(2):126–131, 1957.

Estep, D. A short course on duality, adjoint operators, Green's functions, and a posteriori error analysis. *Lecture Notes*, 2004.

González, J. and Dai, Z. GPyOpt: A Bayesian optimization framework in python (BSD 3 revised license). http://github.com/SheffieldML/GPyOpt, 2016.

Guarnizo, C. and Álvarez, M. A. Impulse response estimation of linear time-invariant systems using convolved gaussian processes and laguerre functions. In Mendoza, M. and Velastín, S. (eds.), *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 281–288, Cham, 2018. Springer International Publishing. ISBN 978-3-319-75193-1.

Guarnizo, C. and Alvarez, M. A. Fast kernel approximations for latent force models and convolved multiple-output Gaussian processes. In *Uncertainty in Artificial Intelligence*, pp. 835–844, 2018.

Higdon, D. Space and space-time modeling using process convolution. In *Quantitative methods for current environmental issues*, pp. 37–56. Springer, 2002.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and De Freitas, N. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

Tobar, F., Bui, T. D., and Turner, R. E. Learning stationary time series using Gaussian processes with nonparametric kernels. In *Advances in Neural Information Processing Systems*, pp. 3501–3509, 2015.