



This is a repository copy of *Adjoint-aided inference of Gaussian process driven differential equations*.

White Rose Research Online URL for this paper:
<https://eprints.whiterose.ac.uk/202736/>

Version: Submitted Version

Preprint:

Gahungu, P., Lanyon, C.W., Alvarez, M.A. et al. (3 more authors) (Submitted: 2022)
Adjoint-aided inference of Gaussian process driven differential equations. [Preprint - arXiv]
(Submitted)

<https://doi.org/10.48550/arxiv.2202.04589>

© 2022 The Author(s). This preprint is made 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.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

Adjoint-aided inference of Gaussian process driven differential equations

Paterne Gahungu¹, Christopher W Lanyon², Mauricio A Álvarez², Engineer Bainomugisha¹, Michael Smith², and Richard D. Wilkinson³

¹College of Computing and Information Sciences, Makerere University

²Department of Computer Science, University of Sheffield

³School 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, after 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 by testing on synthetic data, 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, Schrodinger's, Maxwell's equations etc), as well as systems of linear algebraic equations (such as eigenvalue problems). To fix notation, let

$$\mathcal{L} : \mathcal{U} \rightarrow \mathcal{V}$$

be a linear operator between Banach spaces \mathcal{U} and \mathcal{V} . A prototypical linear system is then of the form

$$\mathcal{L}u = f, \quad (1)$$

where $u \in \mathcal{U}$ is the quantity of interest being modelled. We shall refer to the right hand side, $f \in \mathcal{V}$, as 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} will be a partial differential operator describing the advection and diffusion 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 :

$$z = h(u) + \epsilon. \quad (2)$$

Here $z \in \mathbb{R}^n$ are the observations, h the observation operator, and $\epsilon \in \mathbb{R}^n$ a zero-mean observation error. This is often referred to as the *inverse problem*. 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 zero-mean Gaussian process, with kernel function $k_\psi(\cdot, \cdot)$, in the case where \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 linear operator \mathcal{L} depends (possibly non-linearly) on unknown parameters p . We write \mathcal{L}_p to emphasise this dependence.
3. The observation operator $h : \mathcal{U} \rightarrow \mathbb{R}^n$ is linear (or affine). In the finite dimensional case, $h(u) = Hu$ for some matrix H , but in the infinite dimensional case (i.e., when \mathcal{U} is a space of functions) this includes pointwise evaluation, as well as integral and derivative observations.
4. The observation error ϵ has a Gaussian distribution. This assumption can be relaxed for maximum likelihood estimation.

The full specification of the statistical model is then

$$\mathcal{L}_p u = f \quad (\text{possibly including ICs/BCs}) \quad (3)$$

$$z = h(u) + \epsilon \quad (4)$$

$$f \sim GP(0, k_\psi(\cdot, \cdot)) \quad (5)$$

$$\epsilon \sim \mathcal{N}_n(\mu, \sigma^2 I_n). \quad (6)$$

Our aim is to infer f (and possibly p , u , and ψ) given z either via

- maximum likelihood (ML) estimation, by solving the constrained optimization problem

$$\begin{aligned} \min_{p, f} \quad & (z - h(u))^T (z - h(u)) + \|f\|_V^2 \quad (7) \\ \text{s.t.} \quad & \mathcal{L}_p u = f \end{aligned}$$

- Bayesian inference, where we find

$$\pi(p, f, u | z) \propto \pi(z | u) \pi(u | f, p) \pi(f) \pi(p). \quad (8)$$

Here, $\pi(p)$ and $\pi(f)$ denote the prior distributions for p and f .

To solve either problem numerically is likely to require many solves of the forward problem (1). For example, for ML we may seek a solution using numerical optimization [4], whereas with Bayes, we might use an MCMC scheme [19][30][1] or variational approach [8]. All of these approaches require multiple solves of the forward problem. Our aim is to reduce the number of forward solves required during inference, resulting in lower computational costs.

1.1 Contribution

In this paper we show that implementing an adjoint of the linear system can result in much faster statistical inference for this problem. 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

Source term estimation in differential equation models has been extensively studied, for example, in the field of modelling atmospheric advection-diffusion [35, 6, 32, 27, 36]. with many papers solving the inverse problem using an adjoint approach combined with MCMC to compute the posterior distribution [35, 15, 22, 1]. Of particular relevance is [15] 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 still is still a limiting factor, restricting the extension of the approach to more complex situations such as time-varying pollution sources.

The problem defined by Eqs. (3)-(6) is often referred to as a *latent force model* [2, 3]. [2] showed how the posterior for this model can be computed by using the integral formulation of $\mathcal{L}_p u = f$, $u(x) = \int G_p(x - v) f(v) dv$, where $G(\cdot)$ is the Green's function associated with the differential operator \mathcal{L}_p . 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 $\pi(u | f, p)$ and $\pi(u | p)$ can be computed in closed form. Unfortunately, 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 which can be numerically unstable to compute. Recently, [13] proposed representing f using random Fourier features 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 combining this with a reduced-rank Gaussian process formu-

lation, leading to numerically stable and fast approximations to the posterior distribution.

Other related work includes the stochastic PDE approach of [21], 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, [14] 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 . [31] 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. [16] 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 to the system. There is likely to be benefit in combining these approaches with ours to further accelerate inference in these models.

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

Let \mathcal{U}^* and \mathcal{V}^* denote the dual spaces of \mathcal{U} and \mathcal{V} . We can construct the adjoint to a continuous linear operator \mathcal{L}_p as follows. Let $v^* \in \mathcal{V}^*$ and define $F : \mathcal{U} \rightarrow \mathbb{R}$ by

$$F : u \mapsto v^*(\mathcal{L}_p u). \quad (9)$$

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 $u^* \in \mathcal{U}^*$.

$$\mathcal{L}_p^* : v^* \mapsto u^* = v^* \circ \mathcal{L}_p \quad (10)$$

We call \mathcal{L}_p^* the *adjoint* of \mathcal{L}_p , and \mathcal{L}_p^* is itself a bounded linear operator [11]. By construction, we have that for all $u \in \mathcal{U}$ and $v^* \in \mathcal{V}^*$

$$(\mathcal{L}_p^* v^*)(u) = v^*(\mathcal{L}_p(u)), \quad (11)$$

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

$$\langle \mathcal{L}_p u, v \rangle_{\mathcal{V}} = v^*(\mathcal{L}_p u) = (\mathcal{L}_p^* v^*)(u) \quad (12)$$

$$= \langle u, \mathcal{L}_p^* v \rangle_{\mathcal{U}} \quad (13)$$

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

Generally, the adjoint \mathcal{L}_p^* will be the same type of operator as \mathcal{L}_p (e.g. if \mathcal{L}_p is a differential operator then \mathcal{L}_p^* will be too), so solving an adjoint system of the form $\mathcal{L}_p^* v = g$ will have similar computational complexity as solving $\mathcal{L}_p u = f$. See [11] for an introduction to adjoints, and the experiments section for examples of adjoints.

3.2 Benefits of adjoints

How does the development of an adjoint to a linear system help us perform statistical inference for that system? There are two main advantages. Consider first 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

$$f(\cdot) = \sum_{m=1}^M q_m \phi_m(\cdot). \quad (14)$$

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 length n .

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

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

Then using the bilinear identity (13) we get

$$\begin{aligned} h_i(u) &= \langle h_i, u \rangle = \langle \mathcal{L}_p^* v_i, u \rangle \\ &= \langle v_i, \mathcal{L}_p u \rangle \\ &= \langle v_i, f \rangle, \end{aligned}$$

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: an n -fold increase in computational cost. 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

$$z = \begin{pmatrix} \langle v_1, \phi_1 \rangle & \dots & \langle v_1, \phi_M \rangle \\ \vdots & & \vdots \\ \langle v_n, \phi_1 \rangle & \dots & \langle v_n, \phi_M \rangle \end{pmatrix} \begin{pmatrix} q_1 \\ \vdots \\ q_M \end{pmatrix} + \epsilon \quad (15)$$

$$= \Phi q + \epsilon$$

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 standard linear model. Thus, standard results can be used to compute the least squares/maximum likelihood and Bayesian estimators. For maximum likelihood, the minimum of $S(q) = (z - h(u))^T (z - h(u))$ subject to $\mathcal{L}_p u = f$, is obtained at

$$\hat{q} = (\Phi^T \Phi)^{-1} \Phi^T z$$

with $\text{Var}(\hat{q}) = \sigma^2 (\Phi^T \Phi)^{-1}$ 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

$$q | z \sim \mathcal{N}_M(\mu_n, \Sigma_n) \quad (16)$$

where

$$\mu_n = \Sigma_n \left(\frac{1}{\sigma^2} \Phi^T z + \Sigma_0^{-1} \mu_0 \right), \quad \Sigma_n = \left(\frac{1}{\sigma^2} \Phi^T \Phi + \Sigma_0^{-1} \right)^{-1}. \quad (17)$$

See, e.g., [25]. Note that for readability, we have just presented the approach for Hilbert spaces here, but the idea works in more general Banach spaces.

A second advantage of using adjoints is that the gradient of a cost function depending on the solution u can be computed with respect to parameters p , using the *adjoint sensitivity* approach [7, 23, 34]. For example, consider the unregularized sum of squares/log-likelihood $S(p, q) = (z - h(u))^T (z - h(u))$ from Eq. (7), where u satisfies $\mathcal{L}_p u = f_q$. We've shown how we can estimate $\hat{q}(p) = \arg \max S(p, q)$ or find the posterior $\pi(q | z, p)$, but using the adjoint sensitivity approach also gives us $\frac{\partial}{\partial p} S(p, q)$. This can be used either by a gradient-based optimizer to

learn $\hat{p} = \arg \max S(p, \hat{q}(p))$, or in a variational inference scheme that requires gradients to estimate the variational parameters such as the variational autoencoder [18, 29], or in a Hamiltonian Monte Carlo algorithm [24] to efficiently target the posterior (Eq. 8). We do not explore these approaches in detail here.

3.3 Inference of Gaussian forcing functions

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

$$f(\cdot) \sim GP(m(\cdot), k(\cdot, \cdot)), \quad (18)$$

where $m(\cdot)$ and $k(\cdot, \cdot)$ are the prior mean and covariance functions respectively [28]. Our approach is to use a reduced-rank representation of the GP, as in Eq. (14), derived by truncating an expansion for k of the form

$$k(x, x') = \sum_{m=1}^{\infty} \phi_m(x) \phi_m(x'). \quad (19)$$

There are many possible choices for the basis vectors $\phi_i(\cdot)$, including the Karhunen-Loève (KL) [10], Laplacian [33, 9, 5], and random Fourier feature [26] expansions. The Karhunen Loève basis is formed by finding the spectral expansion of the integral operator defined in Mercer's theorem:

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

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 [28, Section 4.3.2, p116] or numerically [12]. Using the eigenfunctions of T_k gives the L^2 -optimal approximation [20], 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 random Fourier features (RFFs) [26]. 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

$$k(x - x') = \int e^{-iw(x-x')} dp(w). \quad (20)$$

If we use an isotropic exponentiated quadratic kernel,

$$k(x - x') = \tau^2 \exp\left(-\frac{1}{2\lambda^2} (x - x')^T (x - x')\right) \quad (21)$$

then the measure corresponds to a multivariate Gaussian. Using results from [26], we can approximate the Gaussian process with Fourier feature vectors:

$$\phi_i(x) = \sqrt{\frac{2\tau^2}{M}} \cos\left(\frac{1}{\lambda} w_i^T x + b_i\right) \quad (22)$$

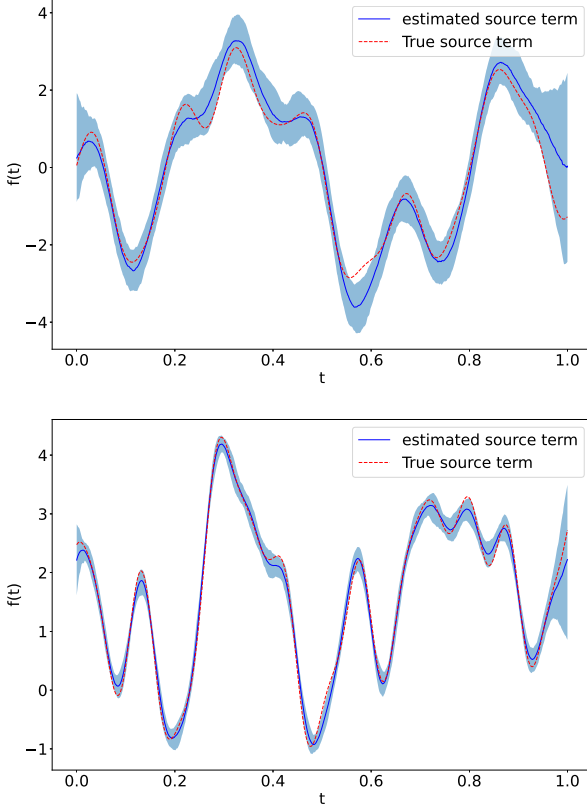


Figure 1: Posterior distribution for the unknown forcing function (shown as 95% credibility interval) with true value overlaid in the case of $n = 50$ (top) and $n = 500$ training points (bottom). A different ground truth is used in each case.

where $w_i \sim N(0, 1)$ and $b_i \sim U(0, 2\pi)$. The extension 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. 19), and so including more terms has a minor effect on overall computational cost.

4 Experiments

4.1 Ordinary differential equation (ODE)

Consider the non-homogeneous linear ODE:

$$p_2 \frac{d^2 u}{dt^2} + p_1 \frac{du}{dt} + p_0 u = f(t) \quad (23)$$

on the domain $[0, T]$ with initial conditions $u(0) = u'(0) = 0$. The right hand side, $f(t)$, is the unknown forcing func-

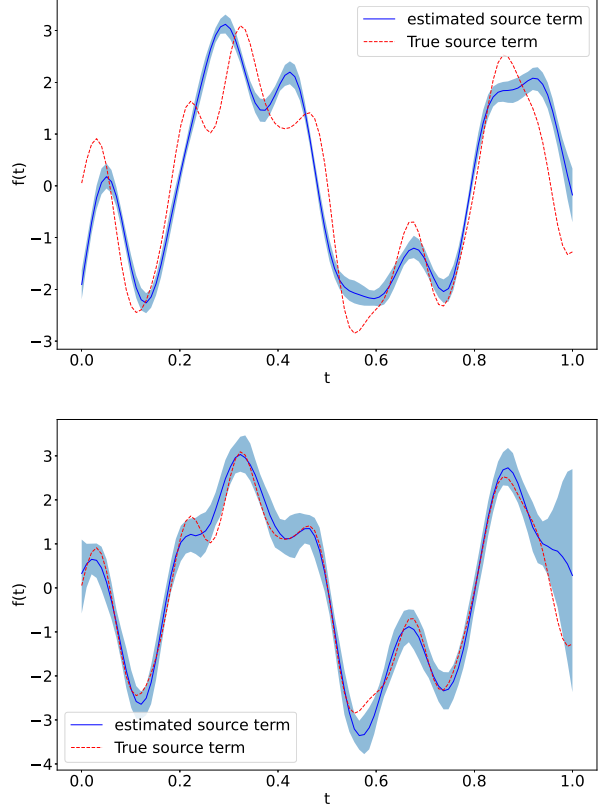


Figure 2: Posterior distribution for the forcing function using $M = 10$ (top) and $M = 150$ (bottom) features in the GP expansion (Eq. 14). True value overlaid in red. The overconfident and wrong posterior when $M = 10$ is a consequence of the model being heavily misspecified in this case.

tion that we wish to estimate, and $p = (p_0, p_1, p_2)^\top$ are parameters in the linear operator. We model f as a zero-mean GP with the exponentiated quadratic covariance function: $f(\cdot) \sim GP(0, k(t, t'))$. We assume observations are obtained as noisy time averages over short time windows:

$$h_i(u) = \int_{t_i}^{t_i + \Delta t} \frac{1}{\Delta t} u(t) dt = \langle u, \tilde{h}_i \rangle \quad (24)$$

where \tilde{h}_i is the indicator function $\tilde{h}_i(t) = \mathbb{I}_{[t_i, t_i + \Delta t]}(t)$. We could include direct measurements of $u(t_i)$ by setting $\tilde{h}_i(t) = \delta(t - t_i)$. To generate synthetic data, we first simulate a realization f from the GP model, solve Eq. (23) for $u(t)$, and then simulate n observations from Eq. (24) 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 (although see the discussion for comments about numerical difficulties). We approximate the

GP using Eq. (14), using 200 RFFs generated using Eq. (22) with $\lambda = \sqrt{0.6}$ and $\tau^2 = 4$. The linear operator in this case is

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

To derive the adjoint operator we use the bilinear identity (Eq. 13), and integrate by parts twice:

$$\begin{aligned} \langle \mathcal{L}u, v \rangle &= \int_0^T \mathcal{L}u(t)v(t)dt = \int_0^T (p_2\ddot{u} + p_1\dot{u} + p_0u)vdt \\ &= [p_2\dot{u}v + p_1uv]_0^T + \int_0^T -p_2\dot{u}\dot{v} - p_1u\dot{v} + p_0uvdt \\ &= [p_2(\dot{u}v + u\dot{v}) + p_1uv]_0^T + \int_0^T p_2u\ddot{v} - p_1u\dot{v} + uvdt \\ &= \int_0^T (p_2\ddot{v} - p_1\dot{v} + v)udt = \langle u, \mathcal{L}^*v \rangle \end{aligned}$$

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

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

Note that rather than an initial condition, the adjoint has a final condition, and so to solve the system we have to integrate it backwards in time from $t = T$ to $t = 0$. A summary of the approach is given in Algorithm 1.

Algorithm 1 Adjoint-aided inference for the ODE model

For $i = 1, \dots, n$:

- Solve the adjoint system $\mathcal{L}_p^* v_i = \tilde{h}_i$ with final condition $v_i(T) = \dot{v}_i(0) = 0$.

For $m = 1, \dots, M$:

- Sample an RFF basis vector $\phi_m(t)$ using Eq. (22)
- Compute $\langle v_i, \phi_m \rangle$ for $1 \leq i \leq n$

Form the matrix Φ , and compute the posterior distribution for q using Eqs. (16) and (17).

Fig. 1 shows the posterior distribution of the forcing function when using $n = 50$ and $n = 500$ data points. As expected, more data results in a more confident posterior. Fig. 2 shows the effect of increasing the number of features in the GP approximation. Note the danger of using too few features: with $M = 10$ the approximation to 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, which can be difficult to spot in real applications.

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:

$$\frac{\partial u}{\partial t} + p_1 \nabla u - \nabla \cdot (p_2 \nabla u) = f \text{ in } \mathcal{X} \times [0, T] \quad (25)$$

with the following initial and boundary conditions:

$$u(x, 0) = 0 \text{ for } x \in \mathcal{X} \quad (26)$$

$$\nabla_n u = 0 \text{ for } x \in \partial \mathcal{X}. \quad (27)$$

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 GP

$$f(x, t) \sim GP(0, k((x, t), (x', t'))) \quad (28)$$

with exponentiated quadratic kernel k (Eq. 21). We use a random Fourier feature approximation to k (Eq. 22), where in this case the random weights are $w_m \sim \mathcal{N}_3(0, I)$ and $b_m \sim U[0, 2\pi]$.

Algorithm 2 Adjoint-aided inference for the PDE model

For $i = 1, \dots, n$:

- Solve the adjoint system $\mathcal{L}_p^* v_i = \tilde{h}_i$ with final and boundary conditions:

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

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

For $m = 1, \dots, M$:

- Sample an RFF basis vector $\phi_m(t)$ using Eq. (22)
- Compute $\langle v_i, \phi_m \rangle$ for $1 \leq i \leq n$.

Form the matrix Φ , and compute the posterior distribution for q using Eqs. (16) and (17).

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

$$z_i = \langle u, \tilde{h}_i \rangle + e_i \quad (29)$$

$$\text{with } \tilde{h}_i = \begin{cases} \frac{1}{|\mathcal{R}_i| \cdot |\mathcal{T}_i|} & \text{if } x \in \mathcal{R}_i \text{ and } t \in \mathcal{T}_i \\ 0 & \text{otherwise.} \end{cases} \quad (30)$$

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

$$\mathcal{L}_p^* v = -\frac{\partial v}{\partial t} - p_1 \nabla v - \nabla \cdot (p_2 \nabla v). \quad (31)$$

This can be derived by integrating by parts as for the ODE, but for partial differential operators, we rely upon Green's theorem to do so; see [17] for details. Our adjoint-aided approach then requires the solution of

$$\mathcal{L}_p^* v_i = \tilde{h}_i \text{ in } \mathcal{X} \times [0, T] \quad (32)$$

with the following initial and boundary conditions:

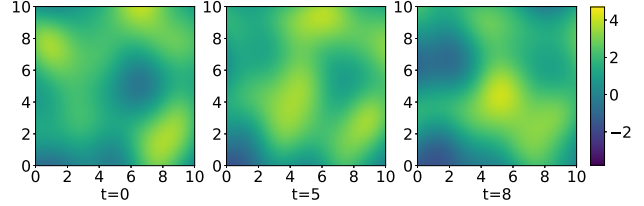
$$\begin{aligned} v(x, T) &= 0 \text{ for } x \in \mathcal{X} \\ p_1 v + p_2 \nabla v &= 0 \text{ for } x \in \partial\Omega \text{ and } t \in [0, T] \end{aligned}$$

for $i = 1, \dots, n$. Inference then proceeds as before. After solving the n adjoint systems, we compute the inner product of these solutions with the random Fourier feature basis vectors to form the matrix Φ (Eq. 15). We can then use Eqs. (16) and (17) to compute the posterior with minimal additional computational cost. The approach is summarized in Algorithm 2.

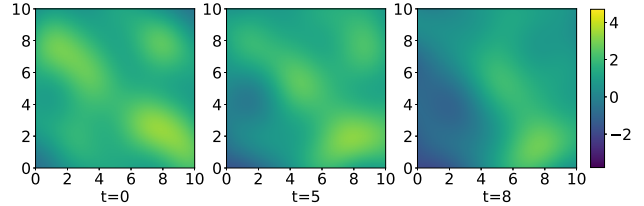
Data was simulated on the spatial domain $\mathcal{X} = [0, 10]^2$ for $t \in [0, 10]$ by first randomly generating a forcing function $f(x, t)$, and then solving the forward problem (Eqs. 25-27) to find $u(x, t)$. We generate n observations using sensors that record averages over short time windows equally spaced across the domain $[0, 10]$, and at the locations shown in Fig. 4. 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 2 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 test the posterior predictive accuracy of the model.

Fig. 3a shows the ground-truth forcing function f (generated from an isotropic GP with an exponentiated quadratic kernel with $\lambda = 2$, $\tau^2 = 2$) as a spatial map at three different time points. Observations were generated for four time windows using both arrays of 4 and 16 sensors at the locations shown in Fig. 4, i.e., a total of either $n = 16$ or $n = 64$ observations. We used $M = 200$ RFFs to infer f using the adjoint-aided approach. The posterior mean for f is shown as spatial maps in Figs. 3b and 3c at 3 different time slices. As expected, more sensors results in better estimates. Figure 4 shows the posterior standard deviation for f in the 4 and 16 sensor cases. Here we used PDE parameters $p_1 = (0.4, 0.4)^\top$ (and $p_2 = 0.01$), so that advection occurs parallel to the $y = x$ line. We can see that standard deviation is greatest *downwind* of the sensors, with a reduction in the variance *upwind*, as expected.

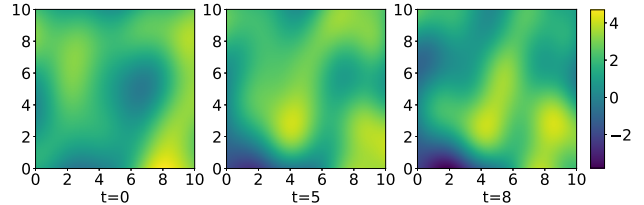
To investigate the effects of varying feature and sensor numbers we performed a posterior predictive check using held-out data and used Monte Carlo estimation to calculate the posterior predictive mean squared error (MSE). Figure 5 gives the MSE as the number of sensors and RFFs vary. As both increase, 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



(a) Ground truth $f(x, t)$ generated with $\lambda = 2$ and $\tau^2 = 2$.



(b) Posterior mean of f using data from 4 sensors, averaged over 4 time windows, using $M = 200$ random Fourier features.



(c) Posterior mean of f using data from 16 sensors, averaged over 4 time windows, using $M = 200$ random Fourier features.

Figure 3: Spatial maps of the forcing function at three different time slices: $f(x, 2)$, $f(x, 5)$, $f(x, 8)$. Top row shows ground truth, middle shows the posterior mean with 4 sensors, bottom row using 16 sensors.

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 evaluation required for the adjoint-aided approach).

Finally, we can infer the remaining parameters (PDE and GP hyperparameters) in a variety of ways. By way of illustration, Figure 6 shows the results from using Bayesian optimization to infer lengthscale in a problem where data are generated with true lengthscale $\lambda = 2$. We used GPyOpt to maximise the negative log-likelihood using the expected improvement acquisition function. In this case, we are easily able to determine the correct lengthscale with relatively minimal computational cost.

5 Discussion

This work was motivated by the problem of inferring (with uncertainty) the distribution of spatially and temporally varying sources of pollution (e.g. across a city), given a small set of (noisy) observations, and a model of the pol-

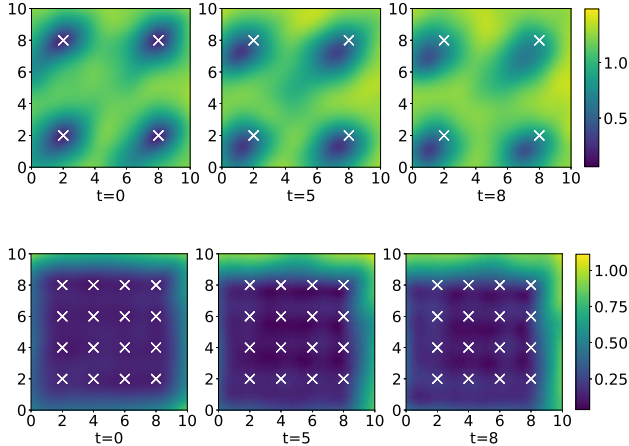


Figure 4: Posterior standard deviation for the examples shown in Fig. 3, given as spatial maps at three time slices. The top row is for the case where 4 sensors are used, and the bottom row for 16 sensors. Sensor locations are shown as white crosses. Advection occurs parallel to the $x = y$ line, so regions of reduced uncertainty are immediately upwind of each sensor.

lution’s atmospheric transport. This is typical of the type of challenge faced throughout the sciences and engineering. Linear systems such as this, still pose computational challenges that make any form of 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. This paper combines the adjoint approach, which allows us to write the problem as a standard linear model, with a linear-sum formulation of a Gaussian process for the forcing term. The result is conjugate Bayesian inference for an unknown GP forcing function, which can vary in both time and space. The computational cost scales linearly¹ in time 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, there is still a large range of problems for which an adjoint may be beneficial, more so if it used to also compute gradients. In our PDE example with relatively few observations, the adjoint-aided approach requires orders of magnitude fewer PDE solves than MCMC. It is also worth noting that the computational tasks in our approach are embarrassingly parallelisable.

One issue that is only briefly touched upon in this paper, is the problem of estimating the non-linear operator param-

¹Specifically, with G grid elements and M features, the algorithm has computational complexity $O(GMn + M^2n + M^3)$.

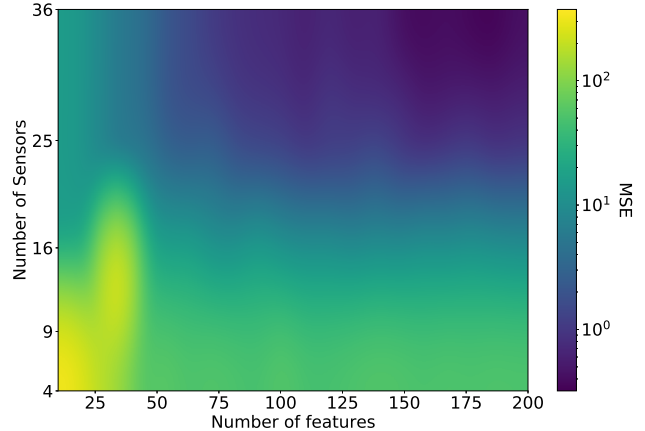


Figure 5: The posterior predictive mean squared error (MSE) as a function of the number of sensors and random fourier features. The ground truth was generated from a GP with $\lambda = 2$ and $\tau^2 = 2$, and using advection-diffusion parameters $p_1 = (0.04, 0.04)$ and $p_2 = 0.01$.

eters p , such as advection and diffusion rates, as well as the GP kernel hyperparameters. Using adjoint sensitivity to estimate gradients with respect to these parameters is a numerically stable alternative to using auto-diff frameworks (such as TensorFlow), which can be unstable when back propagating through long iterative loops. Knowledge of gradients then allows inference for the additional parameters to be performed efficiently within the preferred statistical paradigm. A cautionary note is that when the observation operator \tilde{h} has small support, such as for pointwise evaluation (i.e., $h(u) = u(x, t)$), then care is needed when using numerical methods to solve the adjoint systems, as adaptive step size methods can easily miss small non-zero regions in \tilde{h} .

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.

Acknowledgements

This work is funded by EPSRC/GCRF grant EP/T00343X/1, with additional support provided by GOOGLE.ORG. RW is supported by EPSRC projects EP/P010741/1 and EP/S014985/1.

References

- [1] R. A. Albani, V. V. Albani, H. S. Migon, and A. J. S. Neto. Uncertainty quantification and atmospheric

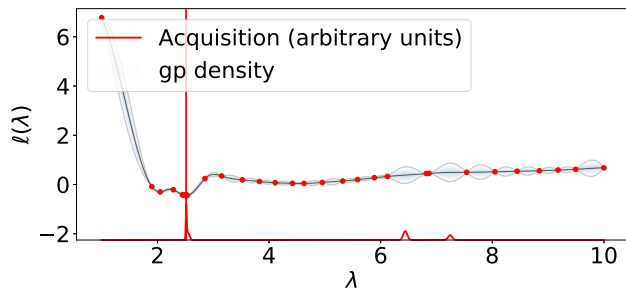


Figure 6: The output of the Bayesian Optimisation algorithm used to infer the value of the GP kernel lengthscale, λ . In this case the true value of λ is 2 and the algorithm found the minimum at $\lambda = 2.52$. This plot shows where the objective function was evaluated and the posterior mean and variance at each point.

source estimation with a discrepancy-based and a state-dependent adaptative MCMC. *Environmental Pollution*, 290:118039, 2021.

- [2] M. Alvarez, D. Luengo, and N. D. Lawrence. Latent force models. In *Artificial Intelligence and Statistics*, pages 9–16. PMLR, 2009.
- [3] M. A. Alvarez, D. Luengo, and N. D. Lawrence. Linear latent force models using Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2693–2705, nov 2013.
- [4] A. Arellano Jr, K. Raeder, J. Anderson, P. Hess, L. Emmons, D. Edwards, G. Pfister, T. Campos, and G. Sachse. 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.
- [5] V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth. Matérn Gaussian processes on Riemannian manifolds. *arXiv preprint arXiv:2006.10160*, 2020.
- [6] M. Borysiewicz, A. Wawrzynczak, and P. Kopka. 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.
- [7] Bradley, A. M. PDE-constrained optimization and the adjoint method. Online tutorial, https://cs.stanford.edu/~ambrad/adjoint_tutorial.pdf, 2010. Accessed: 2022-01-27.
- [8] M. A. Chappell, A. R. Groves, B. Whitcher, and M. W. Woolrich. Variational Bayesian inference for a non-linear forward model. *IEEE Transactions on Signal Processing*, 57(1):223–236, 2008.
- [9] S. Coveney, C. Corrado, C. H. Roney, D. O’Hare, S. E. Williams, M. D. O’Neill, S. A. Niederer, R. H. Clayton, J. E. Oakley, and R. D. Wilkinson. Gaussian process manifold interpolation for probabilistic atrial activation maps and uncertain conduction velocity. *Philosophical Transactions of the Royal Society A*, 378(2173):20190345, 2020.
- [10] P. Deheuvels and G. V. Martynov. 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.
- [11] D. Estep. A short course on duality, adjoint operators, Green’s functions, and a posteriori error analysis. *Lecture Notes*, 2004.
- [12] P. Greengard and M. O’Neil. Efficient reduced-rank methods for Gaussian processes with eigenfunction expansions. *arXiv preprint arXiv:2108.05924*, 2021.
- [13] C. Guarnizo and M. A. Alvarez. Fast kernel approximations for latent force models and convolved multiple-output Gaussian processes. In *Uncertainty in Artificial Intelligence*, pages 835–844, 2018.
- [14] J. Hartikainen and S. Särkkä. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *2010 IEEE international workshop on machine learning for signal processing*, pages 379–384. IEEE, 2010.
- [15] Y. Hwang, H. J. Kim, W. Chang, K. Yeo, and Y. Kim. Bayesian pollution source identification via an inverse physics model. *Computational Statistics & Data Analysis*, 134:76–92, 2019.
- [16] C. Jidling, N. Wahlström, A. Wills, and T. B. Schön. Linearly constrained Gaussian processes. *Advances in Neural Information Processing Systems (NIPS)*, 2017.
- [17] A. Keats, E. Yee, and F.-S. Lien. Bayesian inference for source determination with applications to a complex urban environment. *Atmospheric environment*, 41(3):465–479, 2007.
- [18] D. P. Kingma and M. Welling. Stochastic gradient VB and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, volume 19, page 121, 2014.

- [19] M. Kopacz, D. J. Jacob, D. K. Henze, C. L. Heald, D. G. Streets, and Q. Zhang. 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.
- [20] D. Kosambi. Statistics in function space. In *DD Kosambi*, pages 115–123. Springer, 2016.
- [21] F. Lindgren, H. Rue, and J. Lindström. 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.
- [22] A. K. Luhar, D. M. Etheridge, Z. M. Loh, J. Noonan, D. Spencer, L. Smith, and C. Ong. 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.
- [23] C. C. Margossian. A review of automatic differentiation and its efficient implementation. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 9(4):e1305, 2019.
- [24] R. M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011.
- [25] A. O’Hagan and J. J. Forster. *Kendall’s advanced theory of statistics, volume 2B: Bayesian inference*, volume 2. Arnold, 2004.
- [26] A. Rahimi, B. Recht, et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, page 5. Citeseer, 2007.
- [27] H. Rajaona. *Inférence Bayésienne adaptative pour la reconstruction de source en dispersion atmosphérique*. PhD thesis, Lille 1, 2016.
- [28] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006.
- [29] G. Roeder, P. Grant, A. Phillips, N. Dalchau, and E. Meeds. Efficient amortised Bayesian inference for hierarchical and nonlinear dynamical systems. In *International Conference on Machine Learning*, pages 4445–4455. PMLR, 2019.
- [30] B. Sengupta, K. J. Friston, and W. D. Penny. Gradient-based MCMC samplers for dynamic causal modelling. *NeuroImage*, 125:1107–1118, 2016.
- [31] F. Sigrist, H. R. Künsch, and W. A. Stahel. Stochastic partial differential equation based modelling of large space–time data sets. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, pages 3–33, 2015.
- [32] S. K. Singh and R. Rani. A least-squares inversion technique for identification of a point release: Application to fusion field trials 2007. *Atmospheric environment*, 92:104–117, 2014.
- [33] A. Solin and S. Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 30(2):419–446, 2020.
- [34] I. Yashchuk. 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.
- [35] E. Yee. Theory for reconstruction of an unknown number of contaminant sources using probabilistic inference. *Boundary-layer meteorology*, 127(3):359–394, 2008.
- [36] K. Yeo, Y. Hwang, X. Liu, and J. Kalagnanam. Development of hp-inverse model by using generalized polynomial chaos. *Computer Methods in Applied Mechanics and Engineering*, 347:1–20, 2019.

Supplementary material

PDE Inference Examples

Here we show the effect of the GP lengthscale, λ , on the quality of inference in the PDE problem. Figures 7 to 9 show the ground truth forcing function, $f(x, t)$, simulated concentration, u , and the forcing function inferred with 200 random Fourier features, in the cases where $\lambda = 0.2, 1$ and 3 , respectively. It should be noted that at $t = 0$, u is 0 everywhere, due to the established initial conditions. In the shorter lengthscale cases, the quality of inference is worse than that at $\lambda = 3$ (and at $\lambda = 2$, as shown in main paper). However, the quality of inference in shorter lengthscale cases can be improved by increasing the number of random Fourier features used to perform the adjoint method.

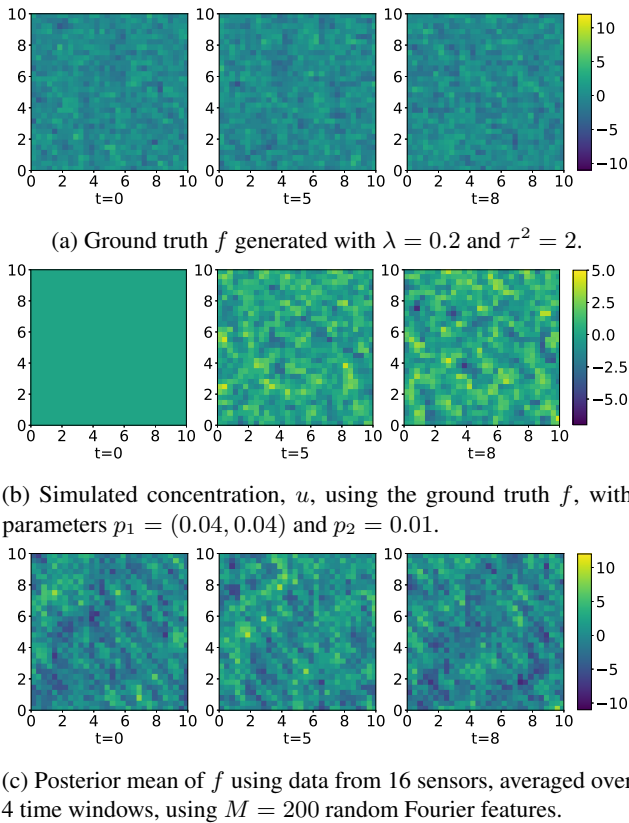


Figure 7: An example of the adjoint inference method applied to a ground truth forcing function, f , with lengthscale $\lambda = 0.2$. The inference was performed using 200 random Fourier features.

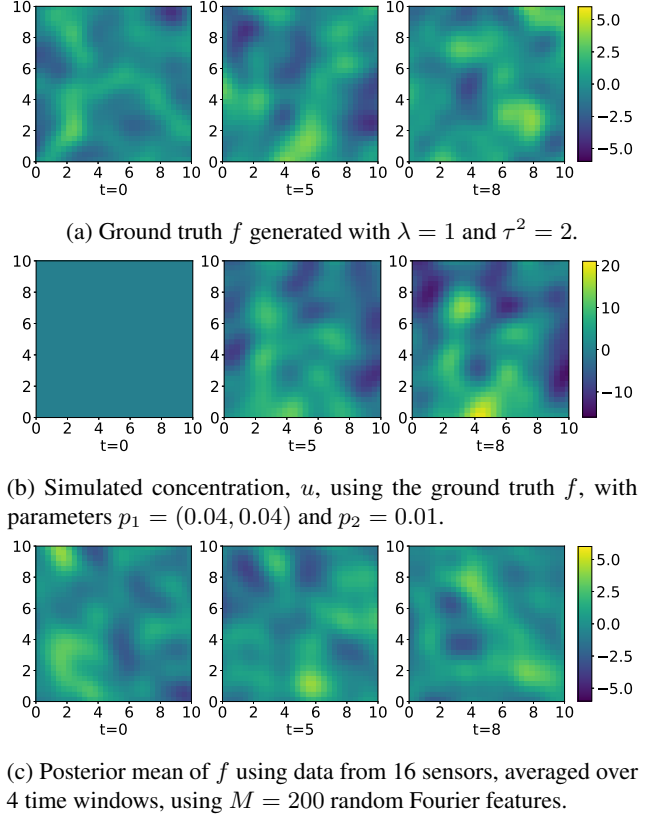
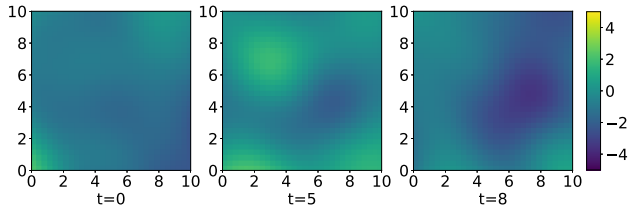


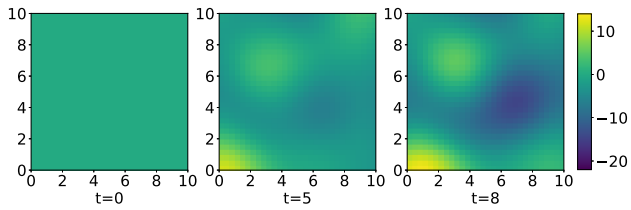
Figure 8: An example of the adjoint inference method applied to a ground truth forcing function, f , with lengthscale $\lambda = 1$. The inference was performed using 200 random Fourier features.

Time complexity

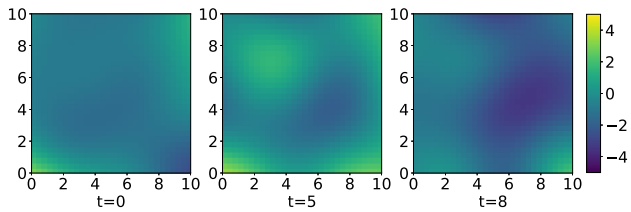
Let G be the number of grid elements, n the number of observations and M the number of features. There are five specific operations our algorithm requires. (1) Solving the n adjoint systems takes $O(Gn)$. (2) Computing each ϕ vector over the grid for each feature takes $O(GM)$. (3) Computing the Φ matrix, takes $O(GMn)$ time (having to dot product each feature computed over the grid with each q associated with each observation). (4) Finding $\Phi^T \Phi$ takes $O(nM^2)$. Finally solving the matrix inverse will take $O(M^3)$. The upshot is an algorithm linear in 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 $O(GMn + M^2n + M^3)$, but the constants associated with each term are important to consider.



(a) Ground truth f generated with $\lambda = 3$ and $\tau^2 = 2$.



(b) Simulated concentration, u , using the ground truth f , with parameters $p_1 = (0.04, 0.04)$ and $p_2 = 0.01$.



(c) Posterior mean of f using data from 16 sensors, averaged over 4 time windows, using $M = 200$ random Fourier features.

Figure 9: An example of the adjoint inference method applied to a ground truth forcing function, f , with lengthscale $\lambda = 3$. The inference was performed using 200 random Fourier features.