How Bayesian Should Bayesian Optimisation Be?

George De Ath
g.de.ath@exeter.ac.uk
Department of Computer Science
University of Exeter
Exeter, United Kingdom

Richard M. Everson
r.m.everson@exeter.ac.uk
Department of Computer Science
University of Exeter
Exeter, United Kingdom

Jonathan E. Fieldsend
j.e.fieldsend@exeter.ac.uk
Department of Computer Science
University of Exeter
Exeter, United Kingdom

ABSTRACT
Bayesian optimisation (BO) uses probabilistic surrogate models – usually Gaussian processes (GPs) – for the optimisation of expensive black-box functions. At each BO iteration, the GP hyperparameters are fit to previously-evaluated data by maximising the marginal likelihood. However, this fails to account for uncertainty in the hyperparameters themselves, leading to overconfident model predictions. This uncertainty can be accounted for by taking the Bayesian approach of marginalising out the model hyperparameters. We investigate whether a fully-Bayesian treatment of the Gaussian process hyperparameters in BO (FBBO) leads to improved optimisation performance. Since an analytic approach is intractable, we compare FBBO using three approximate inference schemes to the maximum likelihood approach, using the Expected Improvement (EI) and Upper Confidence Bound (UCB) acquisition functions paired with ARD and isotropic Matérn kernels, across 15 well-known benchmark problems for 4 observational noise settings. FBBO using EI with an ARD kernel leads to the best performance in the noise-free setting, with much less difference between combinations of BO components when the noise is increased. FBBO leads to over-exploration with UCB, but is not detrimental with EI. Therefore, we recommend that FBBO using EI with an ARD kernel as the default choice for BO.

CCS CONCEPTS
• Theory of computation → Gaussian processes; Mathematical optimization; • Mathematics of computing → Bayesian computation.

KEYWORDS
Bayesian optimisation, Surrogate modelling, Gaussian process, Approximate inference

1 INTRODUCTION
Bayesian optimisation (BO) is a popular sequential approach for optimising costly or time-consuming black-box functions that have no derivative information or closed form [5, 53]. It is a surrogate model-based approach that employs a probabilistic model built with previous evaluations. BO comprises of two main steps, which are repeated until budget exhaustion or convergence. Firstly, a probabilistic surrogate model is constructed, which is typically a Gaussian process (GP) because of their strength in function approximation and uncertainty quantification [15, 45, 52]. Secondly, an acquisition function is optimised to select the next location to expensively evaluate. Acquisition functions combine the surrogate model’s predictions and associated uncertainty to strike a balance between exploiting areas of design space with good predicted values and exploring locations with high uncertainty in their predicted values.

During the first step of a BO iteration, the surrogate model must be learned from the data. The predominant strategy in BO is to find the model hyperparameters that maximise the marginal likelihood, also known as the model evidence. This point-based estimate is known as the maximum likelihood (ML) estimate, or, if we also take into account some prior belief about the model’s hyperparameters, the maximum a posteriori (MAP) estimate. These are normally found via gradient-based optimisation [1, 17, 33]. However, the marginal likelihood landscape may contain multiple optima of similar quality, as well as flat, ridge-like structures on which gradient-based optimisers may get stuck [60]; a common partial remedy is to take the best from multiple optimisations from randomly chosen initial parameters. However, in addition the ML or MAP estimate does not take into account any uncertainty that exists about the true hyperparameters, leading to naturally overconfident predictions.

Viewing this from a Bayesian perspective tells us that we need to marginalise out the hyperparameters of the model; that is, every possible hyperparameter choice should be weighted by how well its corresponding model explains the data. Then, we can use the prediction of these weighted models to take into account the uncertainty in the hyperparameters. In all but the simplest of cases, this requires calculation of an intractable integral, so in practise approximations to the integral are made using methods such as variational inference [25] and Markov Chain Monte Carlo (MCMC) [11, 19, 35]. Several works have performed a fully-Bayesian treatment of the hyperparameters in BO, and some advocate for it to become the prevailing strategy [43, 53]. Yet most works that apply a fully-Bayesian approach, e.g. [2, 21, 59] only use it because it is the correct thing to do, without any more justification. Therefore, in this work, we investigate whether a fully-Bayesian treatment of the surrogate model’s hyperparameters leads to improved performance in BO. Specifically, we investigate the performance of BO using two acquisition functions (Expected Improvement (EI) and Upper Confidence Bound (UCB)), for two different Gaussian process kernel types (isotropic and ARD), and under four different
levels of observation noise. These comparisons are made for the traditional MAP approach and three approximate inference schemes for a fully-Bayesian treatment.

Our main contributions can be summarised as follows:

1. We provide the first empirical study of the effect on BO of a fully-Bayesian treatment of the hyperparameters.
2. We evaluate different combinations of acquisition function, GP kernel type, and inference method on fifteen well-known test functions over a range of dimensions (2 to 10) and for a range of noise levels.
3. We show empirically that using the EI with an ARD kernel and fully-Bayesian inference using MCMC leads to superior BO performance in the noise-free setting.
4. We show that a fully-Bayesian treatment of the hyperparameters always leads to (even more) over-exploration with UCB.

However, for EI a fully Bayesian treatment only increases exploration on higher-dimensional functions with increased observed noise level.

We begin in Section 2 by reviewing BO and how to perform fully-Bayesian BO. In Section 3 we review GPs, paying particular attention to the hyperparameter learning, and follow this up in Section 4 by reviewing the approximate inference schemes used in this work. An extensive experimental evaluation is carried out in Section 5, along with a discussion of the results. We finish with concluding remarks in Section 6.

2 Bayesian Optimisation

Bayesian optimisation (BO), also known as Efficient Global Optimisation, is a surrogate-assisted global search strategy that sequentially samples the problem domain at locations likely to contain the global optimum. It takes into account both the predictions of a probabilistic surrogate model, typically a Gaussian process (GP), and its corresponding uncertainty [24]. It was first proposed by Kushner [29], and improved and popularised by both Mockus et al. [37] and Jones et al. [24]. See [5, 15, 52] for comprehensive reviews of BO. Without loss of generality, we can define the problem of finding a global minimum of an unknown, potentially noise-corrupted objective function $f: \mathbb{R}^d \mapsto \mathbb{R}$ as

$$\min_{x \in X} f(x),$$

defined on a compact domain $X \subset \mathbb{R}^d$. In BO it is assumed that $f$ is black-box, i.e. it has no (known) closed form and no derivative information is available. However, we are able to access the results of its evaluations $f(x)$ at any location $x \in X$. BO is particularly effective in cases where the evaluation budget is limited due to function evaluations being expensive in terms of time and/or money. In this case we wish to optimise $f$ in either as few function evaluations as possible, or as well as possible for a given budget $T$.

Algorithm 1 outlines the standard Bayesian optimisation procedure. It starts (line 1) by generating $S$ initial sample locations $X = \{x_1, \ldots, x_S\}$ with a space-filling sampling, such as Latin hypercube sampling [34]. These are then expensive evaluated with the function $y = \{y_1 \doteq f(x_1), \ldots, y_S \doteq f(x_S)\}$. Then, at each BO iteration, a GP model is usually trained [53] by maximising the log marginal likelihood $\log p(y | X, \theta)$ with respect to the model hyperparameters $\theta$, which are usually parameters of the GP kernel, such as a length scale, and the noise variance assumed to be corrupting the observed value of $f(x)$; see Section 3.3. The marginal likelihood may also be multiplied by a prior probability of the parameters, $p(\theta)$, expressing a priori beliefs about the parameters. Maximisation of $\log p(y | X, \theta)p(\theta)$ obtains the maximum a posteriori (MAP) estimate of the model hyperparameters (line 4). The choice of where to evaluate next in BO is determined by an acquisition function $\alpha(x | y, X, \theta)$, also known as an infill criterion, which balances the exploitation of good regions of the design space found thus far with the exploration regions where the predictive uncertainty is high. Acquisition functions are discussed further in the next section.

### Algorithm 1 Standard Bayesian optimisation.

**Inputs:**
- $S$ : Number of initial samples
- $T$ : Budget on the number of expensive evaluations

**Steps:**
1. $X \leftarrow \text{SpaceFillingSampling}(X, S)$ \quad \rightarrow \text{Initial samples}
2. $y \leftarrow \{y_k \doteq f(x_k)\}_{k=1}^S$ \quad \rightarrow \text{Expensively evaluate all initial samples}
3. for $t = S + 1 \rightarrow T$
4. \quad $\theta \leftarrow \argmax_{\theta} \log[p(y | X, \theta)p(\theta)]$ \quad \rightarrow \text{MAP estimate}
5. \quad $x' \leftarrow \argmax_{x} \alpha(x | y, X, \theta)$ \quad \rightarrow \text{Maximise infill criterion}
6. \quad $f' \leftarrow f(x')$ \quad \rightarrow \text{Expensively evaluate } x'
7. \quad $X \leftarrow X \cup \{x'\}$ \quad \rightarrow \text{Augment training data}
8. \quad $y \leftarrow y \cup \{f'\}$
9. return $D$

2.1 Acquisition Functions

Acquisition functions $\alpha(x | \theta) : \mathbb{R}^d \mapsto \mathbb{R}$ are measures of quality that enable us to decide which location $x \in X$ is the most promising, and thus where we should expend our next expensive evaluation. They are based on the predictive distribution $p(f | x, \theta)$ of the surrogate model, where the dependence on the observed expensive evaluations $(X, y)$ are summarised in $\theta$. Acquisition functions usually depend on both the posterior mean prediction $\mu(x) = \mathbb{E}[p(f | x, \theta)[f(x)]$ and its associated uncertainty captured by the variance $\nu(x) = \mathbb{V}[p(f | x, \theta)]$. Two of the most popular acquisition functions are the Expected Improvement (EI) [24] and Upper Confidence Bound (UCB) [56]. EI measures expected positive improvement over the best function value observed so far $f^*$:

$$\text{aei}(x | \theta) = \mathbb{E}_{p(f | x, \theta)}[\max(f^* - f(x), 0)],$$

which can be expressed analytically as [24]:

$$\text{aei}(x | \theta) = \sqrt{\nu(x)}(s(\Phi(s) + \phi(s)),$$

where $s = (f^* - \mu(x))/\sqrt{\nu(x)}$ is the predicted improvement normalised by its corresponding uncertainty, and $\phi(\cdot)$ and $\Phi(\cdot)$ are the Gaussian probability density and cumulative density functions respectively. EI is known to often be too exploitative, resulting in optimisation runs that can converge prematurely to a local minima.
Various works have tried to curtail this behaviour by increasing the amount of exploration. Berk et al. [3], for example, try to do this by averaging over realisations drawn from surrogate model’s posterior distribution instead of just using the mean prediction. Chen et al. [7], like other authors [13, 55], equate the two terms in EI to exploitation and exploration and up-weight the amount of the latter accordingly. However, as shown by De Ath et al. [9], only certain weight combinations allow for this version of EI to be monotonic in both $\mu(\cdot)$ and $\alpha(\cdot)$; otherwise it can prefer inferior solutions, i.e. with a worse predicted value.

UCB is an optimistic strategy that is the weighted sum of the surrogate model’s posterior mean prediction and its associated uncertainty:

$$a_{\text{UCB}}(x \mid y, X, \theta) = -\left(\mu(x) - \sqrt{\beta_i \sigma(x)}\right), \quad (4)$$

where $\beta_i \geq 0$ is a weight that usually depends on the number of function evaluations performed thus far and that explicitly controls the exploration-exploitation trade-off. Setting $\beta_i$ is non-trivial: too small a value and UCB will get stuck in local optima; too large a value and UCB will become too exploratory, taking too many evaluations to converge. The convergence proofs for UCB of Srinivas et al. [56] rely on a particular schedule for $\beta_i$ in which it increases proportional to the logarithm of $t$, although this scheme has been shown to be over-exploratory for many practical problems [9].

Other acquisition functions have also been proposed such as $\varepsilon$-greedy methods [9], Probability of Improvement [29], Knowledge Gradient [51], and various information-theoretic approaches [20, 21, 48, 59]. However, in this work we focus on EI and UCB due to their popularity.

### 2.2 Fully-Bayesian Bayesian Optimisation

Interestingly, even though fully-Bayesian approaches for Gaussian process (GP) modelling have been proposed in the literature for several decades, e.g. [18, 42], the vast majority of BO works follow Algorithm 1, i.e. they perform a MAP estimate of the GP hyperparameters at each iteration. However, there are some exceptions to this. Osborne [43] developed a Bayesian approach for global optimisation along with a novel acquisition function, and showed that their fully-Bayesian approach outperformed the standard MAP approach. The hugely influential tutorial of Snoek et al. [53] advocates a fully-Bayesian treatment of the GP hyperparameters and shows that it is sometimes superior to MAP estimation. Other works [2, 21, 59] have also performed a fully-Bayesian approach and have used it to illustrate the effectiveness of their proposed acquisition functions, rather than specifically recommending it.

In order to carry out a fully-Bayesian treatment of the hyperparameters in BO, we need to marginalise out the hyperparameters of the surrogate model, i.e. the acquisition function is averaged weighted by the posterior probability of the hyperparameters [53]:

$$\alpha(x \mid y, X) = \int \alpha(x \mid \theta)p(\theta \mid y, X)d\theta, \quad (5)$$

where $p(\theta \mid y, X)$ is the surrogate model’s posterior hyperparameter distribution. The integral appearing in (5) is generally intractable, but it can be approximated via Monte Carlo integration:

$$\alpha(x \mid y, X) \approx \frac{1}{M} \sum_{m=1}^{M} \alpha(x \mid \theta^{(m)}), \quad (6)$$

where $\{\theta^{(1)}, \ldots, \theta^{(M)}\}$ are samples drawn from $p(\theta \mid y, X)$. These samples can be drawn via an approximate inference method such as Hamiltonian Monte Carlo or variational inference, both of which are discussed further in Section 4. We note that the MAP estimate of the hyperparameters can be regarded as approximating $p(\theta \mid y, X)$ by a delta function that places all the posterior probability mass at $\theta_{\text{MAP}} = \text{argmax}_\theta p(\theta \mid y, X)$. In using the estimated integrated acquisition function (6), the uncertainty in the surrogate model hyperparameters is explicitly taken into account and may therefore be expected to lead to acquisition of better locations.

### 3 GAUSSIAN PROCESS SURROGATES

A Gaussian process (GP) defines a prior distribution over functions, such that any finite number of function values are distributed as a multivariate Gaussian [45]. In GP regression we aim to learn a mapping from a collection of inputs $X = \{x_1, \ldots, x_n\}$ to their corresponding outputs $y = \{y_1, \ldots, y_n\}$, where the outputs are often noisy realisations of the underlying function $f(x)$ we wish to model. Assuming that the observations are corrupted with additive Gaussian noise, $y = f + \epsilon$ where $\epsilon \sim N(0, \sigma^2_f)$, the observation model is defined as $p(y \mid f) = N(y \mid f, \sigma^2)$. In GP regression we place a multivariate Gaussian prior over the latent variables $f = \{f_1, \ldots, f_n\}$:

$$p(f \mid X, \theta) \sim N(0, K), \quad (7)$$

with a covariance $K \in \mathbb{R}^{n \times n}$ with $K_{ij} = \kappa(x_i, x_j \mid \theta)$ and associated hyperparameters $\theta$. Here, $\kappa(\cdot, \cdot \mid \theta)$ is a positive semidefinite covariance function modelling the covariance between any pair of locations. For notational simplicity, and without loss of generality, we take the mean function of the GP to be zero; [10] discusses BO performance with different choices of mean function.

#### 3.1 Covariance Functions

The covariance function $\kappa(\cdot, \cdot \mid \theta)$, also known as a kernel (function), encodes prior beliefs about the characteristics of the modelled function, such as its smoothness. Kernels are typically stationary, meaning that they are a function of the distance between the two inputs, i.e. $r = |x - x'|$. One of the most frequently used kernels is the squared exponential (SE) kernel:

$$\kappa_{\text{SE}}(x, x' \mid \theta) = \sigma^2_\text{e} \exp\left(-\frac{r^2}{\ell^2}\right), \quad (8)$$

with parameters $\ell$ and signal variance $\sigma^2_\text{e}$ defining its characteristic length-scale and output-scale respectively. Using a SE kernel for each input dimension with a separate length-scale $\ell_i$ results in the SE automatic relevance determination (ARD) kernel:

$$\kappa_{\text{SE}}(x, x' \mid \theta) = \sigma^2_\text{e} \exp\left(-\sum_{i=1}^{d} \frac{r^2_i}{\ell^2_i}\right), \quad (9)$$

where $r_i = |x_i - x'_i|$. Allowing separate length scales for each dimension allows irrelevant dimensions to be suppressed by placing priors over them which favour large $\ell_i$ [32, 39].
It has been argued [53, 57] that the SE kernel has too strong smoothness assumptions for realistic modelling of physical processes. Popular alternatives include the Matérn family of covariance functions [57]. Here we use the Matérn kernel with $\nu = 5/2$, as recommended by Snoek et al. [53]:

$$
\kappa_{\text{Matérn}}(x, x') \mid \theta = \sigma_u^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left( \sqrt{2\nu} r \right)^\nu K_\nu \left( \sqrt{2\nu} r \right),
$$

where $K_\nu$ is a modified Bessel function and $r^2 = \sum_i d(x_i - x_i')^2 / \ell_i^2$ is the squared distance between $x$ and $x'$ scaled by the length-scales $\ell_i$ which again allows ARD suppression of irrelevant dimensions. Further information on GP kernels can be found in [12, 45].

### 3.2 Making Prediction with GPs

Given some noisy observations $y$ at locations $X$ and a covariance function $\kappa(\cdot | \theta)$, predictions about the underlying function $f$ can be made using the GP model. Assuming a Gaussian noise model, the joint distribution of the observed training values $(X, y)$ function values at a test location $(x', f')$ is

$$
[y | f'] \mid x, \theta, \sigma_e \sim N(0, \kappa(x, x') \mid \theta) + \sigma_e^2 I \kappa(x', x') \mid \theta),
$$

where the elements of the $n$-dimensional vector $\kappa(x, x' \mid \theta)$ are $[\kappa(x, x' \mid \theta)]_i = \kappa(x_i, x'_i \mid \theta)$. Hereafter, the observation noise $\sigma_e$ is incorporated into $\theta$ so that $\theta$ represents both the kernel hyperparameters and the likelihood noise. We also drop the explicit dependence of the kernel on $\theta$ for ease of exposition.

Conditioning the joint distribution (11) on the observations $y$ yields the predictive distribution of $f \mid x', y, X, \theta$ as a Gaussian distribution:

$$
p(f' \mid x', y, X, \theta) \sim N(\mu(x'), \sigma(x')), 
$$

with mean and variance

$$
\mu(x) = \kappa(x, x) \kappa(x, X) + \sigma_e^2 I \kappa(x, x') \mid \theta) \kappa(x', x') \mid \theta)\kappa(x, x) + \sigma_e^2 I \kappa(x, x') \mid \theta)
$$

and

$$
\sigma(x) = \kappa(x, x) - \kappa(x, X) \kappa(x, x') \mid \theta) \kappa(x', x') \mid \theta)\kappa(x, x) + \sigma_e^2 I \kappa(x, x') \mid \theta).
$$

However, before the GP can be used to make predictions the kernel hyperparameters and the observational noise must be inferred.

### 3.3 Learning Hyperparameters

One of the most useful properties of GPs is that we are able to calculate the marginal likelihood $p(y, X \mid \theta) \propto p(y \mid X, \theta)$, otherwise known as the model evidence [31], of data $(X, y)$ for a particular model defined by a set of hyperparameters. The marginal likelihood may be found by direct integration of the product of the likelihood function of the latent variables $f$ and the GP prior. For numerical reasons, the log marginal likelihood is normally used instead:

$$
\log p(y \mid X, \theta) = - \frac{1}{2} y^T \kappa(x, x) + \sigma_e^2 I \kappa(x', x') \mid \theta) y - \frac{1}{2} \log|\kappa(x, x) + \sigma_e^2 I \kappa(x', x') \mid \theta)| - \frac{n}{2} \log 2\pi.
$$

The first term in (15) is the data-fit term, i.e. how well the model predicts the observed targets $y$, while the second corresponds to a complexity penalty that depends only on the magnitude of the covariance function, and the third is a normalisation constant.

The predominant strategy in BO for inferring the hyperparameters is to find the value of $\theta$ that maximises the log marginal likelihood (15). In doing so, we find the maximum likelihood (ML) parameters $\theta_{\text{ML}} = \arg \max_\theta \log p(y \mid X, \theta)$. Predictions can then be made by using $\theta_{\text{ML}}$ in (13) and (14).

However, we often have some prior beliefs about the hyperparameters. For example, if our observations were standardised to have zero mean and unit variance, then it would be extremely unlikely that the likelihood noise would be larger than one because then the model would predict the majority of the observations as noise. These beliefs may be encoded in a prior distribution $p(\theta)$, and can be taken into account in the likelihood evaluation by multiplying the marginal likelihood by the prior distribution:

$$
p(\theta \mid y, X) \propto p(y \mid X, \theta)p(\theta).
$$

In practice, the logarithm of $p(\theta)$ is added to (15) to arrive at the log posterior distribution, which is then maximised to find the maximum a posteriori (MAP) parameters $\theta_{\text{MAP}}$. Note here that if we a priori believe that all hyperparameter configurations are equally likely, then $\theta_{\text{MAP}} = \theta_{\text{ML}}$.

The optimal hyperparameters, $\theta_{\text{MAP}}$ or $\theta_{\text{ML}}$, are normally found by gradient-based optimisation [1, 17, 33] with multiple restarts, using e.g. L-BFGS-B [6]. These restarts are needed because the landscape may contain multiple local maxima [45]. However, the gradient-based optimisation can be sensitive to the starting locations because hyperparameters that are weakly identified can give rise to flat, ridge-like structures [60].

Figure 1 shows four gradient-based optimisation runs on the log marginal likelihood (15) as a function of the length-scale $\ell$ and output-scale $\sigma_o$. Training data was generated by drawing a realisation from a GP with an Matérn 5/2 kernel having hyperparameters $(\ell, \sigma_o, \sigma_e) = (1, 1, 0.1)$. The figure shows the paths taken and final positions (red circles) by gradient based optimisations starting at the locations shown as white circles. Note that $\sigma_e$ was fixed at $\sigma_e = 0.1$. Only two of the optimisation runs ended up at a location close to the maximum, with the other runs failing to get close. The three runs that started on the left-hand side of the figure illustrate the flat, ridge-like structures that can occur – in all three cases the optimiser got stuck in a suboptimal region of almost zero gradient.

Figure 1: Multi-start gradient-based optimisation of the log marginal likelihood (15). Starting locations are shown in white, with optimisation paths shown with red dashed lines and ending locations shown as red circles. Note how only a few runs successfully find the likelihood peak at $(1, 1)$. 

- **Output-scale ($\sigma_o$)**
- **Length-scale ($\ell$)**
- **Log marginal likelihood ($\log p(y | X, \theta)$)**
- **Flat, ridge-like structures observed in BO**

It is important to note that these issues can be addressed by using more sophisticated optimisation algorithms, such as stochastic gradient descent or Bayesian optimisation. However, these methods are computationally more expensive and may not always converge to the global optimum.
As noted above, with the posterior distribution on hand, the averages can be approximated either by simulating draws directly to the GP hyperparameter distribution. In particular, full integral necessary to evaluate \( p_\text{approx}(\theta | D) \) for each iteration of the algorithm, the gradients of the log marginal likelihood of the target density are required for each of the \( L \) steps. In the context of approximate inference for GP regression, this results in the need to invert the kernel matrix \( k(X, X | \theta) \) \( L \) times, hence making HMC a costly procedure to carry out. See [40, 41] for tutorials on HMC and its application to GPs.

Here, we use a self-tuning variant of HMC known as the No-U-Turn Sampler (NUTS) [22], in which both the path length \( L \) and integration step size are automatically tuned. This avoids the additional overhead in manually tuning both parameters each time inference is performed with different data and is thus of particular benefit in BO where the quantity of data grows at each iteration.

## 4.2 Variational Inference

Variational Inference (VI), tries to find a most similar approximate density to a given probability density function [4]. More formally, we first assume a parametrisable family \( q(\theta) \) of approximate densities that captures features of the target density \( p(\theta) \). Then, we try to find \( q^*(\theta) \in Q \) that minimises the Kullback-Leibler (KL) divergence to the target density, \( p(\theta | D) \) for data \( D \):

\[
q^*(\theta) = \arg \min_{q(\theta) \in Q} \text{KL}(q(\theta) \ || \ p(\theta \ | D)).
\]

Finally, we can approximate the posterior with \( q^*(\theta) \) and draw arbitrarily many samples from it to perform approximate inference.
There are many choices for the family $Q$ of approximation, although one of the key ideas behind VI is to choose $Q$ to be expressive enough to model the target density well, but simple enough to allow for efficient optimisation [4]. A few examples include mean-field (MFVI) and full-rank (FRVI) Gaussian approximations, as well as the more recent, expressive, and computationally expensive normalising flows [46].

In this work, we focus on MFVI and FRVI because these have been empirically shown to be suitable for approximating the GP hyperparameter posterior distribution [30], while still being relatively cheap computationally. The mean-field approximation defines $q(\theta)$ as a product of independent densities. Here we choose $q(\theta)$ to be a product of normal densities, one for each hyperparameter:

$$q(\theta; \xi_{mf}) = \mathcal{N}(\theta | m, \text{diag}(\sigma^2)) = \prod_{j=1}^{J} \mathcal{N}(\theta_j | m_j, \sigma_j^2),$$

where $\xi_{mf} = (m_1, \ldots, m_J, \omega_1, \ldots, \omega_J) \in \mathbb{R}^{2J}$ is a vector of unconstrained variational parameters, $\log(\sigma_j^2) = \omega_j$ and $J = |\theta|$ is the number of parameters in the target distribution. Of course, because this is a diagonal approximation to the true posterior, the mean-field approximation will not capture correlation between parameters. In contrast to this, the full-rank Gaussian approximation allows for cross-covariance terms to be modelled explicitly. To ensure that the learned covariance $\Sigma$ is always positive semidefinite, the covariance matrix is written in terms of its Cholesky factorisation, $\Sigma = LL^T$.

This results in an approximating distribution defined as

$$q(\theta; \xi_{fr}) = \mathcal{N}(\theta | m, LL^T),$$

where $\xi_{fr} = (m, L) \in \mathbb{R}^{J(J+1)/2}$ is the vector of unconstrained variational parameters.

VI seeks to minimise the KL divergence between the target density $p(\theta | D)$, and the approximating distribution $q(\theta)$. It can be shown (e.g. [4]) that the KL divergence is minimised by maximising the variational free energy or evidence lower bound (ELBO),

$$\text{ELBO}(q) = \mathbb{E}_q[\log p(\theta | D)] - \mathbb{E}_q[\log q(\theta)].$$

Both terms in ELBO are readily computed, although the model-specific computations required to find the gradient with respect to the parameters $\xi$ for gradient-based optimisation may be cumbersome. More recently, however, the ubiquity of automatic derivative calculations have led to the development of scalable VI algorithms, namely automatic differentiation VI (ADVI) [28]. ADVI first transforms the inference problem into one with unconstrained real-values by, for example, taking the logarithm of parameters that need to be kept strictly positive. It then recasts the gradient of the ELBO as an expectation of $q$ instead, allowing for the use of Monte Carlo methods to perform gradient approximation where needed. Lastly, it reparameterises other gradients in terms of a standard Gaussian, again allowing for the expectation of a gradient to be calculated (easier) instead of the gradient of an expectation (hard to impossible); this is known as the reparameterisation trick [26].

5 EXPERIMENTAL EVALUATION

We now compare the performance of using a fully-Bayesian treatment of the hyperparameters in BO (denoted FBBO), with the standard approach of using the MAP solution (denoted MAP). The EI, and UCB acquisition functions (Section 2.1) are compared for both MAP and FBBO using direct MCMC, MFVI and FRVI across the 15 well-known benchmark functions listed in Table 1. We investigate several scenarios to compare FBBO with MAP. Specifically, we consider the noise-free case, where the function of interest is assumed to not be significantly corrupted by noise; for this we fix $\sigma = 10^{-4}$. We also look at the case where the function is corrupted by additive Gaussian noise for three different levels of noise. Note that, because the functions in Table 1 are noise-free, we add stochastically generated noise to their evaluations to simulate a noisy setting. We modify the functions to have Gaussian additive noise with a standard deviation that is proportional to the range $|f|$ of possible function values. Concretely, we estimate $|f|$ by evaluating $10^5$ Latin hypercube samples (LHS) across the function domain, finding the maximum $f_{\text{max}}$ of these samples, and calculating $|f| = f_{\text{max}} - f_{\text{min}}$, where $f_{\text{min}}$ is the known minimum of the function. Therefore, for a given $\sigma_n$, each function evaluation is a draw from a Gaussian distribution: $y = N(f(x), (\sigma_n|f|)^2) = f(x) + N(0, (\sigma_n|f|)^2)$. We investigate three noise levels, $\sigma_n \in \{0.05, 0.1, 0.2\}$. We also compare FBBO and MAP using ARD and isotropic kernels, i.e. using one length-scale for each dimension or using one length-scale for all dimensions of the problem. One might expect, given the reduction in the number of hyperparameters in the isotropic case, that the hyperparameter posterior distribution would be less complex and therefore less likely to benefit from a fully-Bayesian treatment. Overall, this results in 8 sets of experiments across the test functions: the noise-free setting and three different amounts of noise, repeated for the ARD and isotropic kernels.

A zero-mean GP with a Matérn 5/2 kernel (10) was used for all experiments. At each BO iteration, input variables were rescaled to reside in [0, 1]d, and observations were rescaled to have zero-mean and unit variance prior to GP inference. Relatively uninformative priors were used, based on BoTorch recommendations [1], for the three types of hyperparameters: $t \sim \text{Ga}(3, 6)$, $\sigma_a \sim \text{Ga}(2, 0.15)$, and $\sigma_c \sim \text{Ga}(1.1, 0.05)$, where $\text{Ga}(a, b)$ is a Gamma distribution with concentration and rate parameters $a$ and $b$ respectively. Models were initially trained on $S = 2d$ observations generated by maximin LHS and then optimised for a further $200 - S$ function evaluations. The trade-off $\beta$ between exploitation and exploration in UCB was set using Theorem 1 in [56], which implies that $\beta$ increases logarithmically with $t$. Each optimisation run was repeated 51 times from a different set of LHS samples, and the sets of initial locations were used for all methods to enable statistical comparison. An odd number (51) of repeats were chosen to allow for the calculation of the median fitness value without the need for rounding. For MAP

| Table 1: Benchmark functions and dimensionality (d). |
|-------------------------------|-----------------|
| Name | $d$ |
| Branin | 2 |
| Ackley | 5, 10 |
| Eggholder | 2 |
| Michalewicz | 5, 10 |
| GoldsteinPrice | 2 |
| StyblinskiTang | 5, 7, 10 |
| SixHumpCamel | 2 |
| Hartmann6 | 6 |
| Hartmann3 | 3 |
| Rosenbrock | 7, 10 |

Function formulae can be found at: http://www.sfu.ca/~ssurjano/optimization.html.
estimation, the GP hyperparameters were estimated by maximising 
\( \log[p(y | X, \theta) p(\theta)] \) using a multi-start strategy with L-BFGS-B
[6] and 10 restarts. In all the approximate fully-Bayesian methods
the acquisition functions were averaged over \( M = 256 \) posterior
samples (6). HMC/MCMC sampling inference (Section 4.1) was carried
out using PyMC3 [50], with 4 chains, discarding the first 2048
samples as burn-in and taking every 50th sample. We note that
this is significantly more than previous works, e.g. [21, 59] only
performed inference every 10 BO iterations, using MCMC to draw
\( M = 200 \) samples, discarding only the first 50 of them, and taking
every 3rd sample. In the non-MCMC BO iterations, the authors
reused the latest set of samples. When carrying out variational in-
ference (Section 4.2), optimisation of the ELBO (22) was undertaken
for a maximum of 40000 steps or until convergence. Finally, GP
models were built using GPyTorch [16] and the resulting acquisition
functions were optimised using BoTorch [1]. Full experimental results
are available in the supplementary material, and all code,
initial locations and optimisation runs are available online².

Here, we report performance in terms of the logarithm of the
simple regret \( R_t \), which is the difference between the true minimum
value \( f_{\min} \) and the best seen function evaluation up to iteration \( t \):

\[ \log(R_t) = \log(\min(f_1, \ldots, f_t) - f_{\min}). \]

5.1 Results and Discussion

Each combination of acquisition function (EI and UCB) and kernel
(isotropic and ARD) were evaluated on the test problems shown in
Table 1, for the four different noise levels \((\sigma_n \in \{0, 0.05, 0.1, 0.2\})\).

Figure 3 shows, for the EI and UCB acquisition functions and
noise levels, the number of times BO with each inference method
was the best-performing according to a one-sided, paired Wilcoxon
signed-rank test [27] with Holm-bonferroni [23] correction \((p \geq
0.05)\). As can be seen from the plot, for EI with the isotropic ker-
nel, MAP outperforms the other inference methods. Given that
there are only three hyperparameters \( \theta = (f, \sigma_c, \sigma_o) \) regardless of
the problem dimensionality \( d \), this is not wholly surprising. It is
likely that the hyperparameter posterior distribution (18) quickly
becomes sharply unimodal, particularly given the lack of freedom
in the parameters due the single length-scale. This matches the
observations of Rasmussen and Williams [45], who note that as
the amount of data increases, as it does in BO, that one often finds a
local optimum that is orders of magnitude more probable than any
other, and, indeed, that it becomes more sharply peaked.

Conversely, for EI using an ARD kernel MCMC outperforms
MAP. With ARD there are \( 2 + d \) hyperparameters, and thus much
more freedom to allow for different, but equally likely, explanations
for the data. This corresponds to a much more diffuse, and poten-
tially more multimodal, posterior density. We note that this does
not conflict with Rasmussen and Williams’s observation because
the curse of dimensionality means that much more data would be
required for one mode to become dominant and sharply-peaked.

Figure 4 shows some illustrative convergence plots using EI with
an isotropic (upper) and ARD kernel (lower). Each plot shows the median log simple regret, with shading representing the interquartile range over 51 runs.

Table 1, for the four different noise levels (isotropic and ARD) were evaluated on the test problems shown in

Figure 3: EI and UCB optimisation summary. Bar heights

Figure 4: Illustrative convergence plots for 3 benchmark
problems for EI using an isotropic (upper) and ARD kernel
(lower). Each plot shows the median log simple regret, with

How Bayesian Should Bayesian Optimisation Be? GECCO ’21 Companion, July 10–14, 2021, Lille, France

²https://github.com/georgedeath/how-bayesian-should-BO-be
We investigated how a fully-Bayesian treatment of the Gaussian process hyperparameters affects optimisation performance. With noise-free evaluations, EI with a MCMC inference and an ARD kernel was the best combination of evaluated BO components. However, as the observational noise level increased, there was less to differentiate between the components. In the case of EI, MCMC was found to be more effective with an ARD kernel than an isotropic one. We attribute this to additional flexibility to model the complex and possibly multi-modal hyperparameter posterior that is afforded by a kernel that treats different dimensions with different length scales. MCMC is generally superior to variational methods (MFVI and FRVI) because the marginal posterior is often sufficiently complex that it is poorly modelled with the (necessarily unimodal) mean field or full-rank Gaussian approximations.

We do not recommend the fully-Bayesian approach with UCB because the additional hyperparameter uncertainty leads to even greater exploration with the already over-exploratory UCB acquisition function. However, this is not the case for EI and we, therefore, recommend the fully-Bayesian treatment of the hyperparameters in BO using MCMC because it allows for a principled way to increase exploration without any ad hoc enhancements to EI.

Other important future directions also focus on the modelling ability of the surrogate. Particular aspects are the non-stationarity induced as the optimiser converges [e.g. 54] and improving the modelling of functions with degenerate features, such as discontinuities, using deep GPs [8, 49].

ACKNOWLEDGMENTS
This work was supported by Innovate UK [grant numbers 104400 and 105874]. The authors would like to acknowledge the use of the University of Exeter High-Performance Computing (HPC) facility.
REFERENCES

[1] Maximilian Balandat, Brian Karrier, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. 2020. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In Advances in Neural Information Processing Systems. Curran Associates, Inc., New York, 25324–25335.

[2] Romain Benassi, Julien Bect, and Emmanuel Vazquez. 2011. Robust Bayesian Process-Based Global Optimization Using a Fully Bayesian Expected Improvement Criterion. In Learning and Intelligent Optimization (Lecture Notes in Computer Science). Springer, Berlin, Heidelberg, 176–190.

[3] Julian Berk, Vu Nguyen, Sunil Gupta, Santu Rana, and Svetla Venkatesh. 2019. Exploration Enhanced Expected Improvement for Bayesian Optimization. In Machine Learning and Knowledge Discovery in Databases (Lecture Notes in Computer Science). Springer International Publishing, Cham, Switzerland, 621–637.

[4] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. 2017. Variational Inference: A Review for Statisticians. J. Amer. Statist. Assoc. 112, 518 (2017), 859–877.

[5] Eric Brochu, Vlad M Cora, and Nando De Freitas. 2010. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv:1012.2599

[6] Richard H Byrd, Pieluqhuang Lu, Jorie Nocedal, and Ciyu Zhou. 1995. A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing 16, 5 (1995), 1190–1208.

[7] Zhehui Chen, Simon Mak, and C. F. Jeff Wu. 2021. A hierarchical expected improvement method for Bayesian optimization. arXiv:1911.07285

[8] Andreas Damianou and Neil D. Lawrence. 2013. Deep Gaussian Processes. In Artificial Intelligence and Statistics. PMLR, 207–215.

[9] George De Ath, Richard M. Everson, Alma A. M. Rahat, and Jonathan E. Fieldsend. 2021. Greed is Good: Exploration and Exploitation Trade-off in Bayesian Optimisation. ACM Transactions on Evolutionary Learning and Optimization 1, 1 (2021).

[10] George De Ath, Jonathan E Fieldsend, and Richard M. Everson. 2020. What Do You Mean? The Role of the Mean Function in Bayesian Optimisation. In Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion. Association for Computing Machinery, New York, 1623–1631.

[11] Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. 1987. Hybrid Monte Carlo. Physics Letters B 195, 2 (1987), 216–222.

[12] David Duvenaud. 2014. Automatic model construction with Gaussian processes. Ph.D. Dissertation. University of Cambridge.

[13] Zhiwei Feng, Qingfeng Zhang, Qiangfu Zhang, Qiangang Tang, Tao Yang, and Yang Ma. 2015. A multiobjective optimization based framework to balance the global exploration and local exploitation in expensive optimization. Journal of Global Optimization 61, 4 (2015), 677–694.

[14] Maurizio Filippone, Mingjun Zhong, and Mark Girolami. 2013. A comparative evaluation of stochastic-based inference methods for Gaussian process models. Machine Learning 93, 1 (2013), 93–114.

[15] Peter I Frazier. 2018. A tutorial on Bayesian optimization. arXiv:1807.02811

[16] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. 2018. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In Advances in Neural Information Processing Systems. Curran Associates, Inc., New York, 7576–7586.

[17] GPy. since 2012. GPy: A Gaussian process framework in python. http://github.com/SheffieldML/GPy

[18] Mark S. Handcock and Michael L. Stein. 1993. A Bayesian Analysis of Kriging. J. Amer. Statist. Assoc. 88, 422 (1993), 403–410.

[19] Wilfried K. Hastings. 1970. Monte Carlo sampling methods using Markov chains. Biometrika 57, 1 (1970), 97–109.

[20] Philipp Hennig and Christian J. Schuler. 2012. Entropy search for information-efficient global optimization. The Journal of Machine Learning Research 13, 1 (2012), 1809–1837.

[21] Josef Míguez Hernández-Lobato, Matthew W. Hoffman, and Zoubin Ghahramani. 2014. Predictive entropy search for efficient global optimization of black-box functions. In Advances in Neural Information Processing Systems. MIT Press, Cambridge, Massachusetts, 918–926.

[22] Matthew D. Hoffman and Andrew Gelman. 2014. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. Journal of Machine Learning Research 15, 47 (2014), 1593–1623.

[23] Sture Holm. 1979. A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6, 2 (1979), 65–70.

[24] Donald R. Jones, Matthias Schonlau, and William J. Welch. 1998. Efficient Global Optimization of Expensive Black-Box Functions. Journal of Global Optimization 13, 4 (1998), 455–492.

[25] Michael J. Jordan, Tommi S. Jaakkola, and Lawrence K. Saul. 1999. An Introduction to Variational Methods for Graphical Models. Machine Learning 37, 2 (1999), 183–233.

[26] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. PMLR, 1530–1538.

[27] David J. C. MacKay. 1992. Bayesian methods for adaptive models. Ph.D. Dissertation. California Institute of Technology.

[28] David C. MacKay. 1992. The evidence framework applied to classification networks. Neural Computation 4, 5 (1992), 720–736.

[29] David J. C. MacKay. 1999. Comparison of approximate methods for handling hyperparameters. Neural Computation 11, 5 (1999), 1035–1068.

[30] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equation of State Calculations by Fast Computing Machines. The Journal of Chemical Physics 21, 6 (1953), 1085–1092.

[31] Thomas Minka. 2001. A family of algorithms for approximate Bayesian inference. Ph.D. Dissertation. Massachusetts Institute of Technology.

[32] Jonas Möckus, Vytautas Tiesis, and Antanās Žilinskas. 1978. The application of Bayesian methods for seeking the extremum. Towards Global Optimization 2, 1 (1978), 117–129.

[33] Iain Murray and Ryan P Adams. 2010. Slice sampling covariance hyperparameters of latent Gaussian models. In Advances in neural information processing systems, Vol. 23. Curran Associates, Inc., New York, 1725–1731.

[34] Matthew D. Hoffman and Andrew Gelman. 2014. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. Journal of the American Statistical Association 109, 506 (2014), 776–786.

[35] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. 2015. Taking the human out of the loop: A review of Bayesian optimization. Proc. IEEE 104, 1 (2016), 148–175.

[36] Warren Scott, Peter Frazier, and Warren Powell. 2011. The Correlated Knowledge Gradient for Simulation Optimization of Continuous Parameters using Gaussian Process Regression. SIAM Journal on Optimization 21, 3 (2011), 996–1026.

[37] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. 2016. Taking the human out of the loop: A review of Bayesian optimization. Proc. IEEE 104, 1 (2016), 148–175.

[38] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical Bayesian Optimization of Machine Learning Algorithms. In Advances in Neural Information Processing Systems. Curran Associates, Inc., New York, 2951–2959.

[39] Mohammad Shahriari, Yarin Gal, Tom Duvenaud, Alexander Adamou, Richard P. Adams, and Mark Deisenroth. 2016. Taking the Human Out of the Loop: A Review of Bayesian Optimization. Journal of Machine Learning Research 18, 1 (2017), 699–776.

[40] Jonas Møckus, Vytautas Tiesis, and Antanās Žilinskas. 1978. The application of Bayesian methods for seeking the extremum. Towards Global Optimization 2, 1 (1978), 117–129.
1015–1022.

[57] Michael L. Stein. 1999. *Interpolation of Spatial Data*. Springer New York, New York.

[58] Martin J. Wainwright and Michael I. Jordan. 2008. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning* 1, 1–2 (2008), 1–305.

[59] Zi Wang and Stefanie Jegelka. 2017. Max-value Entropy Search for Efficient Bayesian Optimization. In *International Conference on Machine Learning*. PMLR, 3627–3635.

[60] Jeremy J. Warnes and Brian D. Ripley. 1987. Problems with likelihood estimation of covariance functions of spatial Gaussian processes. *Biometrika* 74, 3 (1987), 640–642.
A INTRODUCTION
In this supplement we include extra results that could not be fit into the main paper due to space constraints. In the following sections, when comparing methods, the best method(s) are determined by whether a method either has the lowest median regret or is statistically indistinguishable from the method with the lowest median regret according to a one-sided, paired Wilcoxon signed-rank test \( p \geq 0.05 \).

B INFERENCE SUMMARIES: MAP VS. MCMC
Here we show the inference summary plots with budgets \( T \in \{50, 100, 150, 200\} \) function evaluations.

Figures 1 and 2 summarise the performance of MAP vs MCMC for each combination of acquisition function and kernel (columns), and test problem (rows).
Fig. 2. MAP vs. MCMC inference summary after $T = 150$ (left) and $T = 200$ (right) function evaluations. The colour of each cell corresponds to whether both inference methods were statistically indistinguishable from one another (white), MAP performed better than MCMC (blue) and MCMC performed better than MAP (red).

C  OPTIMISATION SUMMARY FOR DIFFERING LEVELS OF NOISE

Figure 3 summarises the performance of each combination of acquisition function, inference method and kernel type for each of the four noise settings. As can be seen from the plots, as the noise level increases, EI becomes less dominant.

Fig. 3. Optimisation summary for each level of noise. Bar lengths correspond to the number of times each combination of acquisition function, inference method and kernel type was either the best performing or statistically equal to the best performing combination.

D  RESULTS TABLES

In this section we show the results tables for each of the experiments. The tables show the median log simple regret as well as the median absolute deviation (MAD) from the median, a robust measure
of dispersion. The method with the best (lowest) median regret is shown in dark grey, and those that are statistically indistinguishable from the best method are shown in light grey.

| Method       | Branin (2) | Eggholder (2) | GoldsteinPrice (2) | SixHumpCamel (2) | Hartmann3 (3) |
|--------------|------------|---------------|--------------------|------------------|---------------|
| Median       | MAD        | Median        | MAD                | Median           | MAD           |
| MAP          | 1.05 × 10⁻⁴ | 1.22 × 10⁻⁴   | 6.51 × 10⁻¹       | 4.83 × 10⁻¹     | 4.27 × 10⁻¹   |
| MCMC         | 1.68 × 10⁻⁴ | 2.31 × 10⁻⁴   | 6.51 × 10⁻¹       | 4.32 × 10⁻¹     | 3.21 × 10⁻¹   |
| MFVI         | 6.92 × 10⁻⁵ | 8.19 × 10⁻⁵   | 6.58 × 10⁻¹       | 3.66 × 10⁻¹     | 2.43 × 10⁻¹   |
| FRVI         | 1.74 × 10⁻⁴ | 2.20 × 10⁻⁴   | 6.51 × 10⁻¹       | 3.78 × 10⁻¹     | 6.75 × 10⁻⁵   |

| Method       | Ackley5 (5) | Michalewicz5 (5) | StyblinskiTang5 (5) | Hartmann6 (6) | Rosenbrock7 (7) |
|--------------|-------------|------------------|--------------------|---------------|-----------------|
| Median       | MAD         | Median           | MAD                | Median        | MAD             |
| MAP          | 1.89        | 1.51             | 8.21 × 10⁻¹       | 2.27 × 10⁻¹   | 5.33            |
| MCMC         | 2.08        | 2.03             | 8.31 × 10⁻¹       | 4.07 × 10⁻¹   | 3.63            |
| MFVI         | 2.68        | 2.20             | 5.52 × 10⁻¹       | 3.72          | 2.24            |
| FRVI         | 2.71        | 1.11             | 4.33 × 10⁻¹       | 4.98          | 4.98            |

Table 1. Tabulated results for the EI acquisition function using an isotropic kernel on the noise-free problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.

| Method       | Branin (2) | Eggholder (2) | GoldsteinPrice (2) | SixHumpCamel (2) | Hartmann3 (3) |
|--------------|------------|---------------|--------------------|------------------|---------------|
| Median       | MAD        | Median        | MAD                | Median           | MAD           |
| MAP          | 1.31 × 10⁻¹ | 1.23 × 10⁻¹   | 7.00 × 10⁻¹       | 6.13            | 2.89 × 10⁻¹   |
| MCMC         | 1.25 × 10⁻¹ | 1.70 × 10⁻¹   | 5.47               | 7.26            | 5.99          |
| MFVI         | 8.35 × 10⁻² | 8.30 × 10⁻²   | 6.86 × 10⁻¹       | 3.40            | 3.99          |

Table 2. Tabulated results for the EI acquisition function using an isotropic kernel on the σₙ = 0.05 problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.
| Method      | Branin (2) | Eggholder (2) | GoldsteinPrice (2) | SixHumpCamel (2) | Hartmann3 (3) |
|-------------|------------|---------------|--------------------|------------------|---------------|
| MAP         | 6.57 × 10⁻² | 8.76 × 10⁻² | 6.84 × 10¹ | 2.37 × 10⁻¹ | 5.55 × 10⁻² |
| MCMC        | 1.34 × 10⁻¹ | 9.90 × 10⁻² | 7.11 × 10¹ | 7.75    | 5.66 × 10⁻¹ |
| MFVI        | 9.78 × 10⁻² | 9.41 × 10⁻² | 7.13 × 10¹ | 8.22    | 4.22 × 10⁻² |
| FRVI        | 1.19 × 10⁻¹ | 9.89 × 10⁻² | 6.86 × 10¹ | 1.31 × 10⁻¹ | 5.12 × 10⁻² |

Table 3. Tabulated results for the EI acquisition function using an isotropic kernel on the σ₁ = 0.1 problems.

The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.

| Method      | Ackley5 (5) | Michalewicz5 (5) | StyblinskiTang5 (5) | Hartmann6 (6) | Rosenbrock7 (7) |
|-------------|-------------|-------------------|---------------------|---------------|-----------------|
| MAP         | 8.38        | 2.74              | 1.32               | 6.33 × 10⁻¹ | 4.49 × 10⁻¹ |
| MCMC        | 9.67        | 6.68              | 1.54               | 6.38 × 10⁻¹ | 5.35 × 10⁻¹ |
| MFVI        | 9.12        | 3.75              | 1.95               | 7.21 × 10⁻¹ | 5.94 × 10⁻¹ |
| FRVI        | 9.46        | 6.85              | 1.79               | 5.52 × 10⁻¹ | 5.62 × 10⁻¹ |

| Method      | StyblinskiTang7 (7) | Ackley10 (10) | Michalewicz10 (10) | Rosenbrock10 (10) | StyblinskiTang10 (10) |
|-------------|---------------------|--------------|-------------------|-------------------|-----------------------|
| MAP         | 6.96 × 10¹ | 1.89 × 10¹ | 1.69 × 10¹ | 1.56               | 2.93 × 10⁻¹ |
| MCMC        | 9.68 × 10¹ | 2.36 × 10¹ | 1.89 × 10¹ | 1.25               | 5.25 × 10⁻¹ |
| MFVI        | 9.75 × 10¹ | 2.45 × 10¹ | 1.96 × 10¹ | 7.46 × 10⁻¹ | 4.30 × 10⁻¹ |
| FRVI        | 9.52 × 10¹ | 1.95 × 10¹ | 1.94 × 10¹ | 1.04               | 5.80          |

Table 4. Tabulated results for the EI acquisition function using an isotropic kernel on the σ₁ = 0.2 problems.

The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.
Table 5. Tabulated results for the EI acquisition function using an ARD kernel on the noise-free problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.

| Method | Branin (2) | Eggholder (2) | GoldsteinPrice (2) | SixHumpCamel (2) | Hartmann3 (3) |
|--------|------------|---------------|--------------------|------------------|--------------|
| MAP    | $1.93 \times 10^{-4}$ | $2.24 \times 10^{-4}$ | $6.51 \times 10^1$ | $5.22 \times 10^1$ | $2.17 \times 10^{-1}$ | $2.29 \times 10^{-1}$ | $1.26 \times 10^{-4}$ | $1.65 \times 10^{-4}$ | $9.52 \times 10^{-6}$ | $9.64 \times 10^{-6}$ |
| MCMC   | $1.47 \times 10^{-4}$ | $1.68 \times 10^{-4}$ | $3.19 \times 10^1$ | $4.73 \times 10^1$ | $3.26 \times 10^{-1}$ | $4.32 \times 10^{-1}$ | $1.64 \times 10^{-4}$ | $1.91 \times 10^{-4}$ | $9.37 \times 10^{-6}$ | $1.06 \times 10^{-5}$ |
| MFVI   | $1.04 \times 10^{-4}$ | $1.14 \times 10^{-4}$ | $6.57 \times 10^1$ | $1.09$ | $2.96 \times 10^{-1}$ | $3.46 \times 10^{-1}$ | $1.00 \times 10^{-4}$ | $1.05 \times 10^{-4}$ | $1.41 \times 10^{-5}$ | $1.67 \times 10^{-5}$ |
| FRVI   | $8.98 \times 10^{-5}$ | $1.18 \times 10^{-4}$ | $6.51 \times 10^1$ | $1.37 \times 10^1$ | $3.16 \times 10^{-1}$ | $4.15 \times 10^{-1}$ | $1.85 \times 10^{-4}$ | $2.20 \times 10^{-4}$ | $1.19 \times 10^{-5}$ | $1.35 \times 10^{-5}$ |

Table 6. Tabulated results for the EI acquisition function using an ARD kernel on the $\sigma_0 = 0.05$ problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.

| Method | Branin (2) | Eggholder (2) | GoldsteinPrice (2) | SixHumpCamel (2) | Hartmann3 (3) |
|--------|------------|---------------|--------------------|------------------|--------------|
| MAP    | $3.16 \times 10^{-1}$ | $1.71 \times 10^1$ | $3.59$ | $1.11$ | $5.57$ | $8.15 \times 10^{-1}$ | $1.31 \times 10^{-1}$ | $5.56 \times 10^{2}$ | $7.57 \times 10^{-1}$ | $2.81 \times 10^{-1}$ |
| MCMC   | $2.88 \times 10^{-1}$ | $1.91 \times 10^1$ | $3.52$ | $1.04$ | $2.76$ | $7.34 \times 10^{-1}$ | $1.35 \times 10^{-1}$ | $9.11 \times 10^{2}$ | $8.01 \times 10^{-1}$ | $2.71 \times 10^{-1}$ |
| MFVI   | $3.34 \times 10^{-1}$ | $8.67$ | $3.13$ | $3.96 \times 10^1$ | $3.65$ | $7.43 \times 10^{-1}$ | $1.53 \times 10^{-1}$ | $5.76 \times 10^{2}$ | $7.49 \times 10^{-1}$ | $2.30 \times 10^{-1}$ |
| FRVI   | $2.40 \times 10^{-1}$ | $1.32 \times 10^1$ | $3.19$ | $7.66 \times 10^{-1}$ | $2.79$ | $9.08 \times 10^{-1}$ | $1.14 \times 10^{-1}$ | $4.88 \times 10^{2}$ | $6.78 \times 10^{-1}$ | $1.73 \times 10^{-1}$ |
Table 7. Tabulated results for the EI acquisition function using an ARD kernel on the \( \sigma_n = 0.1 \) problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.

| Method          | Branin (2)  | Eggholder (2) | GoldsteinPrice (2) | SixHumpCamel (2) | Hartmann3 (3) |
|-----------------|-------------|---------------|--------------------|------------------|--------------|
| Map             | Median MAD  | Median MAD    | Median MAD         | Median MAD       | Median MAD   |
| MAP             | 7.92 x 10^{-2} | 7.25 x 10^{-2} | 5.80 x 10^{1}     | 3.41 x 10^{2}   | 7.92         |
| MCMC            | 9.71 x 10^{-2} | 1.13 x 10^{-2} | 6.65 x 10^{1}     | 1.94 x 10^{2}   | 9.68         |
| MFVI            | 7.99 x 10^{-2} | 8.86 x 10^{-2} | 6.85 x 10^{1}     | 5.04 x 10^{2}   | 7.19         |
| FRVI            | 1.11 x 10^{-1} | 1.12 x 10^{-1} | 6.65 x 10^{1}     | 1.94 x 10^{2}   | 7.12         |

| Method          | Ackley5 (5) | Michalewicz5 (5) | StyblinskiTang5 (5) | Hartmann6 (6) | Rosenbrock7 (7) |
|-----------------|-------------|------------------|---------------------|---------------|-----------------|
| Map             | Median MAD  | Median MAD       | Median MAD          | Median MAD    | Median MAD      |
| MAP             | 1.07 x 10^1 | 4.29             | 1.95                | 4.00 x 10^{-1} | 3.07 x 10^1    |
| MCMC            | 1.00 x 10^1 | 3.74             | 1.45                | 3.41 x 10^{-1} | 1.96 x 10^1    |
| MFVI            | 1.13 x 10^1 | 6.27             | 1.75                | 3.71 x 10^{-1} | 5.20 x 10^1    |
| FRVI            | 1.04 x 10^1 | 6.23             | 1.80                | 3.78 x 10^{-1} | 5.44 x 10^1    |

Table 8. Tabulated results for the EI acquisition function using an ARD kernel on the \( \sigma_n = 0.2 \) problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.

| Method          | Branin (2)  | Eggholder (2) | GoldsteinPrice (2) | SixHumpCamel (2) | Hartmann3 (3) |
|-----------------|-------------|---------------|--------------------|------------------|--------------|
| Map             | Median MAD  | Median MAD    | Median MAD         | Median MAD       | Median MAD   |
| MAP             | 7.92 x 10^{-2} | 7.25 x 10^{-2} | 5.80 x 10^{1}     | 3.41 x 10^{2}   | 7.92         |
| MCMC            | 9.71 x 10^{-2} | 1.13 x 10^{-2} | 6.65 x 10^{1}     | 1.94 x 10^{2}   | 9.68         |
| MFVI            | 7.99 x 10^{-2} | 8.86 x 10^{-2} | 6.85 x 10^{1}     | 5.04 x 10^{2}   | 7.19         |
| FRVI            | 1.11 x 10^{-1} | 1.12 x 10^{-1} | 6.65 x 10^{1}     | 1.94 x 10^{2}   | 7.12         |

| Method          | Ackley5 (5) | Michalewicz5 (5) | StyblinskiTang5 (5) | Hartmann6 (6) | Rosenbrock7 (7) |
|-----------------|-------------|------------------|---------------------|---------------|-----------------|
| Map             | Median MAD  | Median MAD       | Median MAD          | Median MAD    | Median MAD      |
| MAP             | 1.07 x 10^1 | 4.29             | 1.95                | 4.00 x 10^{-1} | 3.07 x 10^1    |
| MCMC            | 1.00 x 10^1 | 3.74             | 1.45                | 3.41 x 10^{-1} | 1.96 x 10^1    |
| MFVI            | 1.13 x 10^1 | 6.27             | 1.75                | 3.71 x 10^{-1} | 5.20 x 10^1    |
| FRVI            | 1.04 x 10^1 | 6.23             | 1.80                | 3.78 x 10^{-1} | 5.44 x 10^1    |
| Method | Branin (2) Median | Eggholder (2) Median | GoldsteinPrice (2) Median | SixHumpCamel (2) Median | Hartmann3 (3) Median |
|--------|------------------|----------------------|---------------------------|------------------------|---------------------|
| MAP    | $1.69 \times 10^{-3}$ | $1.72 \times 10^{-3}$ | $9.50 \times 10^{1}$ | $2.96 \times 10^{1}$ | $1.56$ | $1.42$ | $1.02 \times 10^{-3}$ | $1.23 \times 10^{-3}$ | $6.09 \times 10^{-4}$ | $4.03 \times 10^{-4}$ |
| MCMC   | $3.90 \times 10^{-3}$ | $4.75 \times 10^{-3}$ | $1.06 \times 10^{2}$ | $3.48 \times 10^{1}$ | $2.60$ | $1.88$ | $3.93 \times 10^{-3}$ | $3.84 \times 10^{-3}$ | $6.81 \times 10^{-4}$ | $5.78 \times 10^{-4}$ |
| MFVI   | $2.68 \times 10^{-3}$ | $2.71 \times 10^{-3}$ | $1.04 \times 10^{2}$ | $5.15 \times 10^{1}$ | $2.76$ | $2.06$ | $5.36 \times 10^{-3}$ | $4.83 \times 10^{-3}$ | $1.08 \times 10^{-3}$ | $7.36 \times 10^{-4}$ |
| FRVI   | $4.86 \times 10^{-3}$ | $5.21 \times 10^{-3}$ | $8.55 \times 10^{1}$ | $6.97 \times 10^{1}$ | $1.85$ | $1.96$ | $4.86 \times 10^{-3}$ | $3.35 \times 10^{-3}$ | $7.76 \times 10^{-4}$ | $6.42 \times 10^{-4}$ |

| Method | Ackley5 (5) Median | Michalewicz5 (5) Median | StyblinskiTang5 (5) Median | Hartmann6 (6) Median | Rosenbrock7 (7) Median |
|--------|--------------------|-------------------------|---------------------------|---------------------|-----------------------|
| MAP    | $3.74$             | $5.42 \times 10^{-3}$   | $1.15$                    | $2.59 \times 10^{-3}$ | $2.79 \times 10^{-3}$ |
| MCMC   | $4.67$             | $1.02$                  | $2.67 \times 10^{-3}$    | $4.36 \times 10^{-3}$ | $6.22$                |
| MFVI   | $5.08$             | $7.63 \times 10^{-3}$   | $1.42$                    | $4.23 \times 10^{-3}$ | $6.90$                |
| FRVI   | $4.86$             | $9.71 \times 10^{-3}$   | $1.40$                    | $4.06 \times 10^{-3}$ | $8.43$                |

Table 9. Tabulated results for the UCB acquisition function using an isotropic kernel on the noise-free problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.

| Method | Branin (2) Median | Eggholder (2) Median | GoldsteinPrice (2) Median | SixHumpCamel (2) Median | Hartmann3 (3) Median |
|--------|------------------|----------------------|---------------------------|------------------------|---------------------|
| MAP    | $8.33 \times 10^{-2}$ | $9.12 \times 10^{-2}$ | $8.36 \times 10^{1}$ | $3.62 \times 10^{1}$ | $8.09$ | $8.18$ | $2.95 \times 10^{-2}$ | $2.66 \times 10^{-2}$ | $5.48 \times 10^{-3}$ | $3.29 \times 10^{-3}$ |
| MCMC   | $1.23 \times 10^{-1}$ | $1.02 \times 10^{-1}$ | $7.93 \times 10^{1}$ | $6.09 \times 10^{1}$ | $6.20$ | $7.42$ | $4.40 \times 10^{-1}$ | $3.46 \times 10^{-2}$ | $1.30 \times 10^{-2}$ | $8.03 \times 10^{-3}$ |
| MFVI   | $1.09 \times 10^{-1}$ | $8.55 \times 10^{-1}$ | $8.45 \times 10^{1}$ | $3.46 \times 10^{1}$ | $6.67$ | $6.47$ | $4.29 \times 10^{-2}$ | $5.20 \times 10^{-2}$ | $1.38 \times 10^{-2}$ | $1.15 \times 10^{-2}$ |
| FRVI   | $1.03 \times 10^{-1}$ | $1.12 \times 10^{-1}$ | $1.01 \times 10^{2}$ | $2.94 \times 10^{1}$ | $6.70$ | $6.44$ | $5.21 \times 10^{-2}$ | $4.26 \times 10^{-2}$ | $1.45 \times 10^{-2}$ | $9.79 \times 10^{-3}$ |

| Method | Ackley10 (10) Median | Michalewicz10 (10) Median | StyblinskiTang10 (10) Median | Hartmann6 (6) Median | Rosenbrock10 (10) Median |
|--------|----------------------|---------------------------|-----------------------------|---------------------|-------------------------|
| MAP    | $4.23$               | $8.57 \times 10^{-1}$   | $1.52$                      | $4.67 \times 10^{-1}$ | $3.56 \times 10^{1}$  |
| MCMC   | $6.64$               | $1.52$                  | $2.35$                      | $3.48 \times 10^{1}$ | $8.14 \times 10^{1}$  |
| MFVI   | $6.73$               | $1.53$                  | $2.41$                      | $3.28 \times 10^{1}$ | $8.14 \times 10^{1}$  |
| FRVI   | $6.89$               | $1.31$                  | $2.23$                      | $3.56 \times 10^{1}$ | $8.14 \times 10^{1}$  |

Table 10. Tabulated results for the UCB acquisition function using an isotropic kernel on the $\sigma_p = 0.05$ problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.
### Table 11. Tabulated results for the UCB acquisition function using an isotropic kernel on the $\sigma_\text{init} = 0.1$ problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.

| Method       | Branin (2) | Eggholder (2) | GoldsteinPrice (2) | SixHumpCamel (2) | Hartmann3 (3) |
|--------------|------------|---------------|--------------------|------------------|---------------|
| MAP          | 6.99 × 10^{-2} | 7.82 × 10^{-1} | 9.60 × 10^{-1} | 2.10 × 10^{0}   | 6.49          |
| MCMC         | 1.03 × 10^{-1} | 1.02 × 10^{-2} | 2.74 × 10^{-1} | 8.04             | 8.45          |
| MFVI         | 1.24 × 10^{-1} | 1.30 × 10^{-1} | 8.84 × 10^{-1} | 3.39 × 10^{-1}  | 8.84          |
| FRVI         | 1.18 × 10^{-1} | 1.01 × 10^{-2} | 1.01 × 10^{-1} | 3.15 × 10^{-1}  | 4.91          |

### Table 12. Tabulated results for the UCB acquisition function using an isotropic kernel on the $\sigma_\text{init} = 0.2$ problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.

| Method       | Branin (2) | Eggholder (2) | GoldsteinPrice (2) | SixHumpCamel (2) | Hartmann3 (3) |
|--------------|------------|---------------|--------------------|------------------|---------------|
| MAP          | 7.83 × 10^{-1} | 5.03 × 10^{-1} | 4.51 × 10^{0}    | 1.69 × 10^{1}   | 1.96 × 10^{-1} |
| MCMC         | 1.85 × 10^{-1} | 2.32          | 4.83 × 10^{-1}   | 8.01 × 10^{-1}  | 2.09 × 10^{-1} |
| MFVI         | 1.93 × 10^{-1} | 1.51          | 3.99 × 10^{-1}   | 7.80 × 10^{-1}  | 2.25 × 10^{-1} |
| FRVI         | 1.85 × 10^{-1} | 2.46          | 4.06 × 10^{-1}   | 8.01 × 10^{-1}  | 2.17 × 10^{-1} |

| Method       | Michalewicz5 (5) | StyblinskiTang5 (5) | Hartmann6 (6) | Rosenbrock7 (7) |
|--------------|-------------------|---------------------|---------------|-----------------|
| MAP          | 7.94 × 10^{-1} | 1.81 × 10^{-1} | 5.39          | 4.79 × 10^{-1} |
| MCMC         | 1.17 × 10^{-1} | 1.94 × 10^{-1} | 2.01 × 10^{-1} | 6.47 × 10^{-1} |
| MFVI         | 1.17 × 10^{-1} | 2.13 × 10^{-1} | 2.01 × 10^{-1} | 6.47 × 10^{-1} |
| FRVI         | 1.17 × 10^{-1} | 2.12 × 10^{-1} | 2.02 × 10^{-1} | 6.47 × 10^{-1} |

| Method       | Ackley10 (10) | Michalewicz10 (10) | StyblinskiTang10 (10) |
|--------------|----------------|---------------------|----------------------|
| MAP          | 7.84 × 10^{-1} | 1.81 × 10^{-1} | 5.39          |
| MCMC         | 1.17 × 10^{-1} | 1.94 × 10^{-1} | 2.01 × 10^{-1} |
| MFVI         | 1.17 × 10^{-1} | 2.13 × 10^{-1} | 2.01 × 10^{-1} |
| FRVI         | 1.17 × 10^{-1} | 2.12 × 10^{-1} | 2.02 × 10^{-1} |

| Method       | Branin (2) | Eggholder (2) | GoldsteinPrice (2) | SixHumpCamel (2) | Hartmann3 (3) |
|--------------|------------|---------------|--------------------|------------------|---------------|
| MAP          | 2.48 × 10^{-1} | 3.04 × 10^{-1} | 9.33 × 10^{-1} | 3.61 × 10^{-1}  | 5.75          |
| MCMC         | 3.07 × 10^{-1} | 2.65 × 10^{-1} | 7.92 × 10^{-1} | 5.58 × 10^{-1}  | 3.84          |
| MFVI         | 3.77 × 10^{-1} | 3.43 × 10^{-1} | 7.28 × 10^{-1} | 5.10 × 10^{-1}  | 5.73          |
| FRVI         | 4.42 × 10^{-1} | 4.44 × 10^{-1} | 8.51 × 10^{-1} | 4.13 × 10^{-1}  | 2.75          |

| Method       | Ackley10 (10) | Michalewicz10 (10) | StyblinskiTang10 (10) |
|--------------|----------------|---------------------|----------------------|
| MAP          | 1.08 × 10^{0}  | 6.89             | 4.77 × 10^{-1} | 4.54 × 10^{-1}  |
| MCMC         | 1.93 × 10^{1}  | 1.02             | 2.60          | 4.37 × 10^{-1}  |
| MFVI         | 1.93 × 10^{1}  | 1.22             | 2.65          | 2.64 × 10^{-1}  |
| FRVI         | 1.93 × 10^{1}  | 1.15             | 2.74           | 3.22 × 10^{-1}  |

| Method       | StyblinskiTang7 (7) | Ackley10 (10) | Michalewicz10 (10) | Rosenbrock10 (10) |
|--------------|---------------------|----------------|---------------------|-------------------|
| MAP          | 7.41 × 10^{-1} | 2.02 × 10^{-1} | 1.90 × 10^{-1} | 1.15          |
| MCMC         | 1.28 × 10^{-2} | 2.19 × 10^{-1} | 2.01 × 10^{-1} | 5.79 × 10^{-1}  |
| MFVI         | 1.26 × 10^{-2} | 1.99 × 10^{-1} | 2.01 × 10^{-1} | 5.77 × 10^{-1}  |
| FRVI         | 1.26 × 10^{-2} | 2.08 × 10^{-1} | 1.99 × 10^{-1} | 6.47 × 10^{-1}  |
| Method        | Branin (2)          | Eggholder (2)         | GoldsteinPrice (2)     | SixHumpCamel (2)      | Hartmann3 (3)       |
|--------------|---------------------|-----------------------|------------------------|-----------------------|---------------------|
|              | Median (10^−3)      | Median (10^−3)        | Median (10^−3)         | Median (10^−3)        | Median (10^−3)      |
| MAP          | 2.98 × 10^−3        | 3.12 × 10^−3         | 4.04 × 10^2            | 1.88 × 10^3           | 3.29 × 10^3         |
| MCMC         | 6.10 × 10^−3        | 7.76 × 10^−3         | 9.44 × 10^1            | 4.78 × 10^2           | 2.17 × 10^3         |
| MFVI         | 7.48 × 10^−3        | 9.78 × 10^−3         | 9.26 × 10^1            | 5.88 × 10^2           | 1.32 × 10^3         |
| FRVI         | 5.87 × 10^−3        | 7.57 × 10^−3         | 1.05 × 10^2            | 4.96 × 10^1           | 1.92 × 10^3         |

Table 13. Tabulated results for the UCB acquisition function using an ARD kernel on the noise-free problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.

| Method        | StylbinskiTang (7) | Ackley10 (10)         | Michalewicz10 (10)     | Rosenbrock10 (10)     | StylbinskiTang10 (10) |
|--------------|---------------------|-----------------------|------------------------|-----------------------|-----------------------|
|              | Median (10^1)       | Median (10^1)         | Median (10^4)          | Median (10^1)         | Median (10^1)         |
| MAP          | 7.12 × 10^1         | 1.34 × 10^2          | 6.05 × 10^1            | 4.83 × 10^2           | 5.42 × 10^2           |
| MCMC         | 1.06 × 10^2         | 2.09 × 10^1          | 7.99 × 10^−1           | 5.41 × 10^−1         | 4.20 × 10^−1          |
| MFVI         | 1.15 × 10^2         | 1.62 × 10^1          | 8.36 × 10^−1           | 5.48 × 10^−1         | 7.28 × 10^−1          |
| FRVI         | 1.19 × 10^2         | 1.94 × 10^1          | 8.17 × 10^−1           | 5.14 × 10^−1         | 7.15 × 10^−1          |

Table 14. Tabulated results for the UCB acquisition function using an ARD kernel on the σ_0 = 0.05 problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left- and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.
Method  | Branin (2)  | Eggholder (2)  | GoldsteinPrice (2)  | SixHumpCamel (2)  | Hartmann3 (3)  
|--------|------------|----------------|---------------------|-------------------|----------------|
| MAP    | $3.28 \times 10^{-2}$ | $3.58 \times 10^{-2}$ | $8.31 \times 10^{1}$ | $3.21 \times 10^{4}$ | $5.56$ | $5.21$ | $3.07 \times 10^{-2}$ | $3.72 \times 10^{-2}$ | $9.47 \times 10^{-3}$ | $3.06 \times 10^{-3}$ |
| MCMC   | $6.85 \times 10^{-2}$ | $7.48 \times 10^{-2}$ | $8.35 \times 10^{1}$ | $4.09 \times 10^{4}$ | $5.60$ | $5.91$ | $4.04 \times 10^{-2}$ | $4.53 \times 10^{-2}$ | $1.72 \times 10^{-2}$ | $7.04 \times 10^{-3}$ |
| MFVI   | $7.32 \times 10^{-2}$ | $8.50 \times 10^{-2}$ | $9.11 \times 10^{1}$ | $5.66 \times 10^{4}$ | $5.77$ | $4.79$ | $4.08 \times 10^{-2}$ | $4.86 \times 10^{-2}$ | $1.45 \times 10^{-2}$ | $5.84 \times 10^{-3}$ |
| FRVI   | $7.25 \times 10^{-2}$ | $7.16 \times 10^{-2}$ | $1.01 \times 10^{5}$ | $4.46 \times 10^{1}$ | $5.22$ | $6.22$ | $5.07 \times 10^{-2}$ | $4.34 \times 10^{-2}$ | $1.44 \times 10^{-2}$ | $7.01 \times 10^{-3}$ |

Table 15. Tabulated results for the UCB acquisition function using an ARD kernel on the $\sigma_n = 0.1$ problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.

Method  | Branin (2)  | Eggholder (2)  | GoldsteinPrice (2)  | SixHumpCamel (2)  | Hartmann3 (3)  
|--------|------------|----------------|---------------------|-------------------|----------------|
| MAP    | $7.26$ | $2.26$ | $1.97$ | $4.00 \times 10^{-1}$ | $3.44 \times 10^{1}$ | $1.31 \times 10^{4}$ | $2.08 \times 10^{-1}$ | $6.69 \times 10^{-2}$ | $7.24 \times 10^{-1}$ | $5.46 \times 10^{-1}$ |
| MCMC   | $1.70 \times 10^{1}$ | $3.59$ | $2.26$ | $3.44 \times 10^{-1}$ | $6.91 \times 10^{1}$ | $1.99 \times 10^{4}$ | $2.78 \times 10^{-1}$ | $6.03 \times 10^{-2}$ | $9.33 \times 10^{-1}$ | $7.78 \times 10^{-1}$ |
| MFVI   | $1.66 \times 10^{1}$ | $4.48$ | $2.20$ | $3.13 \times 10^{-1}$ | $7.12 \times 10^{1}$ | $2.01 \times 10^{4}$ | $3.00 \times 10^{-1}$ | $8.37 \times 10^{-2}$ | $7.22 \times 10^{-1}$ | $7.69 \times 10^{-1}$ |
| FRVI   | $1.71 \times 10^{1}$ | $3.42$ | $2.18$ | $3.27 \times 10^{-1}$ | $7.46 \times 10^{1}$ | $1.98 \times 10^{4}$ | $2.72 \times 10^{-1}$ | $3.84 \times 10^{-2}$ | $7.33 \times 10^{-1}$ | $7.13 \times 10^{-1}$ |

Table 16. Tabulated results for the UCB acquisition function using an ARD kernel on the $\sigma_n = 0.2$ problems. The median log simple regret and the median absolute deviation from the median (MAD) is shown in the left and right-hand columns respectively. The method with the lowest median performance is shown in dark grey, with those that are statistically equivalent shown in light grey.

### E CONVERGENCE AND DISTANCE PLOTS

The figures in this section show the convergence and distance plots for each combination of acquisition function, kernel type and level of function noise. The convergence plots show the
median log simple regret, with shading representing the interquartile range over 51 runs, and the dashed vertical line indicating the end of the initial LHS phase. The distance plots show the normalised Euclidean distance between consecutively selected locations over the optimisation run. For each $d$-dimensional problem, distances are normalised by the largest possible distance possible, i.e. $\sqrt{d}$, so that distances reside in $[0,1]$.

Fig. 4. Convergence (upper) and distance (lower) plots for the EI acquisition function with an isotropic kernel on the noise-free problems.
Fig. 5. Convergence (upper) and distance (lower) plots for the EI acquisition function with an isotropic kernel on the $\sigma_n = 0.05$ problems.
Fig. 6. Convergence (upper) and distance (lower) plots for the EI acquisition function with an isotropic kernel on the $\sigma_n = 0.1$ problems.
Fig. 7. Convergence (upper) and distance (lower) plots for the EI acquisition function with an isotropic kernel on the $\sigma_n = 0.2$ problems.
Fig. 8. Convergence (upper) and distance (lower) plots for the EI acquisition function with an ARD kernel on the noise-free problems.
Fig. 9. Convergence (upper) and distance (lower) plots for the EI acquisition function with an ARD kernel on the $\sigma_n = 0.05$ problems.
Fig. 10. Convergence (upper) and distance (lower) plots for the EI acquisition function with an ARD kernel on the $\sigma_n = 0.1$ problems.
Fig. 11. Convergence (upper) and distance (lower) plots for the EI acquisition function with an ARD kernel on the $\sigma_n = 0.2$ problems.
Fig. 12. Convergence (upper) and distance (lower) plots for the UCB acquisition function with an isotropic kernel on the noise-free problems.
Fig. 13. Convergence (upper) and distance (lower) plots for the UCB acquisition function with an isotropic kernel on the $\sigma_n = 0.05$ problems.
Fig. 14. Convergence (upper) and distance (lower) plots for the UCB acquisition function with an isotropic kernel on the $\sigma_n = 0.1$ problems.
Fig. 15. Convergence (upper) and distance (lower) plots for the UCB acquisition function with an isotropic kernel on the $\sigma_n = 0.2$ problems.
Fig. 16. Convergence (upper) and distance (lower) plots for the UCB acquisition function with an ARD kernel on the noise-free problems.
Fig. 17. Convergence (upper) and distance (lower) plots for the UCB acquisition function with an ARD kernel on the $\sigma_n = 0.05$ problems.
Fig. 18. Convergence (upper) and distance (lower) plots for the UCB acquisition function with an ARD kernel on the $\sigma_n = 0.1$ problems.
Fig. 19. Convergence (upper) and distance (lower) plots for the UCB acquisition function with an ARD kernel on the $\sigma_n = 0.2$ problems.

REFERENCES

[1] Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6, 2 (1979), 65–70.

[2] Joshua D. Knowles, Lothar Thiele, and Eckart Zitzler. 2006. *A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers*. Technical Report TIK214. Computer Engineering and Networks Laboratory, ETH Zurich, Zurich, Switzerland.