Bayesian Analysis (2023) **18**, Number 1, pp. 133–163

Bayesian Experimental Design Without Posterior Calculations: An Adversarial Approach

Dennis Prangle[*], Sophie Harbisher[†], and Colin S. Gillespie[‡]

Abstract. Most computational approaches to Bayesian experimental design require making posterior calculations repeatedly for a large number of potential designs and/or simulated datasets. This can be expensive and prohibit scaling up these methods to models with many parameters, or designs with many unknowns to select. We introduce an efficient alternative approach without posterior calculations, based on optimising the expected trace of the Fisher information, as discussed by Walker (2016). We illustrate drawbacks of this approach, including lack of invariance to reparameterisation and encouraging designs in which one parameter combination is inferred accurately but not any others. We show these can be avoided by using an adversarial approach: the experimenter must select their design while a critic attempts to select the least favourable parameterisation. We present theoretical properties of this approach and show it can be used with gradient based optimisation methods to find designs efficiently in practice.

Keywords: automatic differentiation, game theory, geostatistical regression, Hyvärinen score, minimax optimisation, pharmacokinetics.

1 Introduction

Selecting a good design for an experiment can be crucial to extracting useful information and controlling costs. Applications include medical interventions (Amzal et al., 2006), epidemic modelling (Cook et al., 2008), pharmacokinetics (Ryan et al., 2014; Overstall and Woods, 2017) and ecology (Gillespie and Boys, 2019). In modern applications it is increasingly feasible to take a large number of measurements – e.g. placing sensors (Krause et al., 2009) or making observations in a numerical integration problem (Oates et al., 2020) – or make other complex designs – e.g. selecting a time series of chemical input levels to a synthetic biology experiment (Bandiera et al., 2018). Therefore finding high dimensional designs is an increasingly relevant task.

We focus on the Bayesian approach to optimal experimental design, which takes into account existing knowledge and uncertainty about the process being studied before the experiment is undertaken. In this framework an *experimenter* must select a design. They then receive some utility based on the outcome of the experiment. The aim is to select the design optimising expected utility given the experimenter's prior beliefs.

[*] School of Mathematics, University of Bristol, UK, dennis.prangle@bristol.ac.uk

[†] School of Mathematics Statistics and Physics, Newcastle University, UK, s.harbisher@newcastle.ac.uk

[‡] School of Mathematics Statistics and Physics, Newcastle University, UK, colin.gillespie@newcastle.ac.uk

https://doi.org/10.1214/22-BA1306

Most utility functions in Bayesian experimental design require posterior calculations, such as evaluating the evidence. Optimisation of the design requires these calculations to be repeated for a large number of potential designs and simulated datasets. This can be expensive and prohibit scaling up these methods to models with many parameters, or designs with many unknowns to select. Some sophisticated approaches have been developed, including converting the optimisation problem into Markov chain Monte Carlo on the joint space of designs, parameters, and observations (Müller, 1999), and using variational inference to learn approximate surrogate posteriors (Foster et al., 2019, 2020), but these still effectively solve inference problems and hence remain costly. (We comment on other related approaches in Section 1.2.)

This paper presents approaches which avoid the need for posterior inference. First we consider a utility function based on the trace of the Fisher information, which is often available in a simple closed form. Walker (2016) presents an information theoretic justification for this utility. One contribution of this paper is to show that it also emerges naturally from the decision theoretic framework of Bernardo (1979). Our derivation is based on judging the quality of a parameter estimate through the Hyvärinen score (Hyvärinen, 2005), rather than through the logarithmic score as in Bernardo (1979). We demonstrate that it is straightforward to optimise the resulting expected utility using recent developments in stochastic gradient optimisation (Kingma and Ba, 2015) and automatic differentiation (Baydin et al., 2017). Compared to existing methods, this approach is fast and scales easily to higher dimensional designs. However, a drawback is that the method can converge to poor local maxima. We show how a second stage of optimisation similar to that of Overstall and Woods (2017) can often be used to find the overall optimal design.

A more fundamental limitation of the above approach is that it sometimes produces poor designs in practice e.g. requiring all observations to occur at a single time point. We provide an explanation: optimising this expected utility encourages designs giving accurate inference of one linear combination of the parameters but not necessarily others. Furthermore this utility is not invariant to reparameterisation: this can alter which parameter combinations are most rewarding to infer accurately.

We address both these issues by introducing an adversarial approach. We propose a game theoretic framework in which, as before, the experimenter chooses a design to optimise their expected utility. Now there is also a *critic* who selects a linear transformation of the parameters. We investigate the optimal designs in this framework under the game theoretic solution concept of subgame perfect equilibrium, and prove they are invariant to reparameterisation. The presence of the critic also encourages designs not to neglect the posterior accuracy of any parameter combination: if one did then the critic could choose a parameterisation concentrating on this weakness. We show it is possible to find optimal designs in this game theoretic framework using generic *gradient descent ascent* methods, which have been much studied recently in the machine learning literature (e.g. Heusel et al., 2017; Jin et al., 2020).

Below, we summarise our contributions in Section 1.1 and related work in Section 1.2. In the remainder of the paper, Section 2 presents background on Bayesian experimental design. Section 3 presents results on the decision theoretic framework of Bernardo (1979), and Section 4, on our proposed game theoretic extension. Proofs and other technical material are presented in the supplementary material (Prangle et al., 2022). Section 5 discusses details of gradient based optimisation for both approaches.

We illustrate our methods on a simple Poisson model where optimal designs can be derived analytically (Section 6). We provide a detailed simulation study on a pharmacokinetic model (Section 7), showing our adversarial approach produces a sensible design and is at least 10 times faster than competing methods. We also present a geostatistical regression example, where hundreds of design choices can be optimised in under a minute (Section 8). Code for these examples is available at https://github.com/dennisprangle/AdversarialDesignCode. All examples were run on a desktop PC with 12 CPU cores. Our conclusion, Section 9, summarises our findings and recommendations to implement our methods. It also discusses limitations of our work, and future research directions to address these, including a discussion of the intractable Fisher information case (detailed further in supplementary material).

1.1 Contributions

Our main contribution is a faster approach to Bayesian experimental design using the Fisher information. Where the Fisher information is available in closed form, our approach outperforms existing methods (Section 7) and scales easily to designs with hundreds of design choices (Section 8).

In addition, we contribute by extending the decision theoretic framework for Bayesian experimental design of Bernardo (1979) to allow other proper scoring rules in addition to logarithmic score (Section 3.1). We also show that using the Hyvärinen score in the decision theoretic framework results in a utility based on the trace of the Fisher information, referred to as Fisher information gain or FIG in the paper (Section 3.3). This provides a decision-theoretic justification to a measure that was previously suggested by Walker (2016) based on information theoretic arguments. We explore the limitations of FIG (Section 3.4) and address them by extending the decision theoretic approach to a game theoretic approach, and show the relevant solution concept gives more intuitively informative designs (Section 4). We also show that the game theoretic solution can easily be obtained in practice using gradient-based minimax optimisation.

Finally, we provide new insight into two long-standing questions in Bayesian experimental design. Firstly: what counts as a Bayesian utility function? (See Sections 3.4 and 4.4 under the heading "Bayesian justification".) Secondly: which of several possible Bayesian generalisations of D-optimality should be preferred? (See Section 4.4 under the heading "Link to D-optimality".)

1.2 Related work

As discussed above, our theoretical contribution builds on Bernardo (1979)'s decision theoretic framework for Bayesian experimental design, and also on Walker (2016)'s justification for the use of a utility based on the trace of the Fisher information. We also discuss Overstall (2022)'s comments on this utility later.

Gradient-based optimisation methods for experimental design have been explored previously. Pronzato and Walter (1985) optimise the expected determinant of the Fisher information using analytically derived gradients. Huan and Marzouk (2013, 2014) optimise expected Shannon information using gradients (either derived analytically or based on finite differences) for a biased numerical approximation to the utility. Foster et al. (2019) use a first stage of variational inference to learn an approximate posterior, and then use this to produce a surrogate expected utility function which is optimised by gradient-based methods in a second stage.

Foster et al. (2020) and Kleinegesse and Gutmann (2020) propose jointly optimising a design and some tuning parameters defining a lower bound on the expected Shannon information gain utility (this utility is discussed in Section 2.4). This can be framed as a single optimisation problem, allowing for easy implementation using stochastic gradient or Bayesian optimisation methods. Producing a tight lower bound is equivalent to, or closely related to, being able to perform exact Bayesian inference for the optimal design.

Overstall and Woods (2017) propose an alternative coordinate ascent approach which loops over the components of the design, updating each in turn. To perform an update, designs are selected from the one dimensional search space (in which only the current component is updated) and Monte Carlo estimates of expected utility calculated. A Gaussian process is fitted to the expected utility estimates and used to propose an improved value for the design component under consideration. This is accepted or rejected based on a Bayesian test of whether it improves expected utility, using a large number of simulations under the current and proposed designs.

Many other algorithms have been proposed for Bayesian experimental design. An influential method of Müller (1999) performs optimal design using Markov chain Monte Carlo. Amzal et al., 2006 and Kück et al., 2006 extend this approach to use sequential Monte Carlo. In other approaches, Ryan et al. (2014) look at high dimensional designs with a low dimensional parametric form, Price et al. (2018) use evolutionary algorithms, and Gillespie and Boys (2019) search a discrete grid of designs.

2 Background

Optimal experimental design concerns the following problem. An experimenter must select a design τ. The experiment produces data y with likelihood $f(y|\theta;\tau)$, where θ is a vector of parameters. The goal is to select the design which optimises some notion of the quality of the experiment, typically based on its informativeness and its cost. We mostly, but not exclusively, concentrate on the case where $\tau \in \mathcal{T} \subseteq \mathbb{R}^d$ and $\theta \in \Theta \subseteq \mathbb{R}^p$ for some closed sets (under the Euclidean metric) $\mathcal{T}$ and Θ, and where y is a vector of n observations. Designs τ of this form often represent times or locations for measurements to be taken. In this case τ can be seen as a set of *design points*, $\tau_1, \tau_2, \dots, \tau_d$.

This section reviews relevant details of the statistical background. First Sections 2.1 and 2.2 give some necessary definitions on Bayesian statistics and Fisher information. Section 2.3 describes Bayesian experimental design, which is based on maximising expected utility. Finally, Section 2.4 summarises some common utility functions.

2.1 Bayesian framework

We work in the Bayesian framework and introduce a prior density $\pi(\theta)$ for θ. We will often make use of the posterior density and the prior predictive density (or *evidence*) for y. In our setting both depend on the experimental design τ,

$$\pi(\theta|y;\tau) = \pi(\theta) f(y|\theta;\tau)/\pi(y;\tau), \tag{1}$$

$$\pi(y;\tau) = \mathbb{E}_{\theta\sim\pi(\theta)}[f(y|\theta;\tau)]. \tag{2}$$

The prior and model define a joint density,[1]

$$\pi(\theta, y;\tau) = \pi(\theta) f(y|\theta;\tau) = \pi(\theta|y;\tau)\pi(y;\tau). \tag{3}$$

2.2 Fisher information

We will make frequent use of the Fisher information matrix (FIM) for θ,

$$\mathcal{I}_\theta(\theta;\tau) = \mathbb{E}_{y\sim f(y|\theta;\tau)}[u(y,\theta;\tau)u(y,\theta;\tau)^T], \tag{4}$$

which is based on the score function,

$$u(y,\theta;\tau) = \nabla_\theta \log f(y|\theta;\tau). \tag{5}$$

The subscript in $\mathcal{I}_\theta$ denotes which variable is used for differentiation in (5). Often this is obvious from the context so the subscript will be dropped. We will also sometimes omit the dependence on τ where this is not relevant. We will focus on models where u and $\mathcal{I}$ are well defined.

Examples The FIM for exponential family models is based on the variance of the sufficient statistics (see e.g. Lehmann and Casella, 2006) which is often available in closed form. Two examples we use in this paper are:

- The Poisson distribution, $y \sim Poisson(\phi)$. Here $\mathcal{I}(\phi) = 1/\phi$.
- The multivariate normal distribution with known variance, $y \sim N(\mu, \Sigma)$. Here $\mathcal{I}(\mu) = \Sigma^{-1}$.

Reparameterisation Consider a probability model with parameters θ, and a function $\phi(\theta)$ producing an alternative vector of parameters ϕ (which may be shorter or longer than θ). Let $J(\phi)$ be the Jacobian i.e. the matrix whose row i column j entry is $\frac{\partial\phi_i}{\partial\theta_j}$. Then it follows from (4) that

$$\mathcal{I}_\theta(\theta) = J(\phi)^T \mathcal{I}_\phi(\phi) J(\phi). \tag{6}$$

An application of this result is to the model $y \sim N(x(\theta,\tau),\Sigma)$. Using (6),

$$\mathcal{I}_\theta(\theta) = J(x)^T \mathcal{I}_x(x) J(x) = J(x)^T \Sigma^{-1} J(x). \tag{7}$$

[1] For discrete y a density with respect to a product of Lebesgue and counting measures can be used.

2.3 Bayesian experimental design

The Bayesian approach to experimental design involves selecting a function $\mathcal{U} = \mathcal{U}(\tau, \theta, y)$, giving the utility of choosing design τ given observations y and true parameters θ. (As we shall see, many choices of $\mathcal{U}$ do not depend on all these possible inputs.) We try to maximise the *expected utility* of τ i.e. the prior predictive utility

$$\mathcal{J}(\tau) = \mathbb{E}_{(\theta,y)\sim\pi(\theta,y;\tau)}[\mathcal{U}(\tau, \theta, y)]. \tag{8}$$

See Chaloner and Verdinelli (1995), Atkinson et al. (2007) and Ryan et al. (2016) for comprehensive surveys of Bayesian experimental design.

2.4 Utility functions

Ideally a utility function could be specified for each application, perhaps by eliciting preferences over different (τ, θ, y) combinations from the experimenter (e.g. Wolfson et al., 1996). However this is rarely feasible in practice. Instead several generic choices of utility have been proposed.

Shannon information gain (SIG) This is a popular and well-motivated utility choice,

$$\mathcal{U}_{\text{SIG}}(\tau, \theta, y) = \log \pi(\theta|y; \tau) - \log \pi(\theta), \tag{9}$$

introduced by Lindley (1956), which is particularly relevant later in this paper. Designs maximising the expectation of $\mathcal{U}_{\text{SIG}}$ have an appealing information theoretic interpretation: they maximise expected reduction in Shannon entropy from prior to posterior. Furthermore, Bernardo (1979) gave a decision theoretic derivation of $\mathcal{U}_{\text{SIG}}$ which we recap in Section 3. A helpful property of $\mathcal{U}_{\text{SIG}}$ is that it is *reparameterisation invariant* i.e. unchanged under a bijective transformation of θ. Thus the resulting designs are not affected by the choice of parameterisation.

In practice, optimising the expectation of (9) is complicated by the need to evaluate the posterior density. However, using (1) the utility can be rewritten as

$$\mathcal{U}_{\text{SIG}}(\tau, \theta, y) = \log f(y|\theta; \tau) - \log \pi(y; \tau). \tag{10}$$

Optimisation now requires evaluation of a posterior summary: the *log evidence* $\log \pi(y; \tau)$. A common SIG estimate replaces $\pi(y; \tau)$ in (10) with a Monte Carlo approximation

$$\hat{\pi}(y; \tau) = \frac{1}{L}\sum_{\ell=1}^{L} f(y|\theta^{(\ell)}; \tau), \tag{11}$$

where $\theta^{(\ell)}$ are independent prior samples. A typical choice of L is 1000 (Overstall and Woods, 2017), which makes each utility evaluation somewhat computationally expensive. Furthermore, a biased estimate of $\mathcal{U}_{\text{SIG}}$ is produced. More efficient approaches are possible, as outlined in Section 1.2. However the need for log evidence estimation remains a source of computational expense and approximation error.

Other utilities and Bayesian justification Many alternative utility functions to $\mathcal{U}_{\text{SIG}}$ are also used in practice e.g. posterior precision and mean squared error between the posterior mean and true parameters. See Chaloner and Verdinelli (1995) and Ryan et al. (2016) for detailed reviews. Like SIG, both utilities just mentioned are also based on posterior calculations. Indeed Ryan et al. (2016) argue that for a utility to be "fully Bayesian", it must be a functional of the posterior, and other utilities, such as scalar summaries of the FIM, are "pseudo-Bayesian". (Throughout we use this definition of "pseudo-Bayesian", but note some authors use the term differently e.g. Overstall, 2022.) One contribution of this paper is to instead use a decision theoretic justification for which utilities to use in a Bayesian framework, which, surprisingly, provides support to some apparently pseudo-Bayesian utilities (see discussion in Section 3.4), and then to further develop this into a game theoretic approach.

3 Theory: decision theory approach

This section explores a decision theoretic approach to underlie Bayesian experimental design. Section 3.1 presents the framework, which is taken from Bernardo (1979). Section 3.2 describes some background on *scoring rules*. Section 3.3 presents a novel theoretical result showing how the framework supports maximisation of expected utility for various classes of utility function derived from scoring rules. This includes Shannon information gain, as shown by Bernardo (1979), but also the trace of the FIM, as proposed by Walker (2016). We conclude in Section 3.4 by discussing advantages and disadvantages of the latter utility, and how these reveal limitations of the decision theory framework, motivating our modification in Section 4.

3.1 Decision theoretic framework

Bernardo (1979) proposed the following decision theoretic framework for Bayesian experimental design. The experimenter selects a design, and then nature generates parameters θ from the prior $\pi(\theta)$ and observations y from the likelihood $f(y|\theta;\tau)$. The experimenter only observes y and must now choose a, a density for θ, receiving reward $\mathcal{R}(a,\theta)$. In this section and Section 4 we will assume that:

A1 The reward is the negative of a *strictly proper scoring rule*, as defined in Section 3.2.

Throughout this section $\mathcal{S}(a,\theta) = -\mathcal{R}(a,\theta)$ denotes the scoring rule.

3.2 Scoring rules

A *scoring rule* $\mathcal{S}(q,\theta)$ measures the quality of a distribution – in this paper represented by its density $q(\theta)$ – to model an uncertain quantity, given a realised value θ. Low scores represent a good match. A scoring rule is *strictly proper* if, given any $p(\theta)$, $\mathbb{E}_{\theta\sim p(\theta)}[\mathcal{S}(q,\theta)]$ is uniquely minimised by $q = p$. For more background on scoring rules see for example Gneiting and Raftery (2007) and Parry et al. (2012).

Given a scoring rule, two related quantities are

$$\begin{aligned} \mathcal{H}[p(\theta)] &= \mathbb{E}_{\theta\sim p(\theta)}[\mathcal{S}(p(\theta),\theta)], && \text{(entropy of } p\text{)} \\ \mathcal{D}[p(\theta),q(\theta)] &= \mathbb{E}_{\theta\sim p(\theta)}[\mathcal{S}(q(\theta),\theta)-\mathcal{S}(p(\theta),\theta)]. && \text{(divergence from } p \text{ to } q\text{)} \end{aligned}$$

Supplementary Section A gives details of the entropy and divergence for two strictly proper scoring rules which will be used below: logarithmic score and Hyvärinen score (Hyvärinen, 2005). Hyvärinen score uses only the derivatives of $\log q(\theta)$, so it can be calculated from unnormalised densities.

3.3 Results

Result 1 is a general result characterising solutions of the decision theoretic framework. It uses the following extra assumption:

A2 Both $\mathbb{E}_{\theta\sim\pi(\theta)}[\,|S(\pi(\theta),\theta)|\,]$ and, for any τ, $\mathbb{E}_{(\theta,y)\sim\pi(\theta,y;\tau)}[\,|S(\pi(\theta|y;\tau),\theta)|\,]$ are finite.

Result 1. *Assume A1 and A2. Then the following are equivalent, in the sense of sharing the same set of optimal designs:*

1. *The experimenter acts to maximise their expected reward.*
2. *The experimenter selects τ to maximise the expectation, with respect to $\pi(\theta,y;\tau)$, of any of the following utilities:*

$$\begin{aligned} \mathcal{U}_{entropy} &= -\mathcal{H}[\pi(\theta|y;\tau)], \\ \mathcal{U}_{entropy\ diff} &= \mathcal{H}[\pi(\theta)] - \mathcal{H}[\pi(\theta|y;\tau)], \\ \mathcal{U}_{divergence} &= \mathcal{D}[\pi(\theta|y;\tau),\pi(\theta)]. \end{aligned}$$

(Note that arguments of utilities are omitted in this section to simplify notation.)

The next result provides further equivalent utility choices for particular scoring rules.

Result 2. *For the logarithmic scoring rule, and assuming A2, maximising the expectation with respect to $\pi(\theta,y;\tau)$ of either $\mathcal{U}_{divergence}$ or $\mathcal{U}_{SIG}$, defined in (9), gives the same set of optimal designs.*

For the Hyvärinen scoring rule the same is true for $\mathcal{U}_{divergence}$ and both of

$$\mathcal{U}_{FIG} = ||\nabla_\tau \log f(y|\theta;\tau)||^2, \quad \mathcal{U}_{trace} = \operatorname{tr}\mathcal{I}(\theta;\tau),$$

assuming regularity condition A5,[2] *defined in supplementary Section A.*

Here $||\theta||$ represents the L_2 norm i.e. $||x|| = \sqrt{x^T x}$.

The first part of Result 2 provides a decision theoretic derivation of the Shannon information gain utility. The argument used is essentially the same as that of Bernardo (1979). In an analogy to this, we refer to $\mathcal{U}_{\mathrm{FIG}}$ in the second part of the result as

[2] A5 implies A2, and also contains some other conditions.

Fisher information gain.[3] We note that Walker (2016) proved directly that $\mathcal{U}_{\text{trace}}$ and the corresponding $\mathcal{U}_{\text{entropy diff}}$ and $\mathcal{U}_{\text{divergence}}$ all have the same expectations up to an additive constant, and were therefore equivalent when used in experimental design. The second part of Result 2 shows that the same conclusion arises from a decision theoretic approach based on the Hyvärinen score.

3.4 Fisher information gain properties

Result 2 supports maximising the expectation of $\mathcal{U}_{\text{FIG}}$, or equivalently of $\mathcal{U}_{\text{trace}}$,

$$\mathcal{J}_{\text{FIG}}(\tau) = \mathbb{E}_{\theta\sim\pi(\theta)}[\operatorname{tr}\mathcal{I}(\theta;\tau)] = \operatorname{tr}\bar{\mathcal{I}}(\tau), \tag{12}$$

where $\bar{\mathcal{I}}(\tau) = \mathbb{E}_{\theta\sim\pi(\theta)}[\mathcal{I}(\theta;\tau)]$. We refer to this as the FIG approach to experimental design, and here we discuss its properties. We will see that despite computational advantages it has several undesirable properties, illustrated later in our examples. Section 4 addresses these issues by generalising the decision theoretic framework.

Bayesian justification Since the definition of $\mathcal{U}_{\text{trace}}$ does not involve y, it is not a functional of the posterior, and therefore is pseudo-Bayesian under the terminology of Ryan et al. (2016) (discussed at the end of Section 2.4.) However, as pointed out by Walker (2016), $\mathcal{J}_{\text{FIG}}$ also results from using utilities which *are* functionals of the posterior – e.g. $\mathcal{U}_{\text{divergence}}$ in our notation – so it can be regarded as fully Bayesian. This shows the divide between pseudo-Bayesian and fully Bayesian approaches can be hard to clearly define, and motivates using a decision (/game) theoretic approach to do so.

Computational advantages The FIM is often available in a closed form which can easily be evaluated, and allows easy evaluation of gradients. Then optimisation of $\mathcal{J}_{\text{FIG}}(\tau)$ is straightforward using standard stochastic optimisation methods, as described in Section 5. In particular, the objective does not involve calculating the log evidence, which creates optimisation difficulties for $\mathcal{J}_{\text{SIG}}$, or any other use of explicit posterior inference. The case where evaluation of the FIM is more complicated is discussed in supplementary Section G.

Lack of reparameterisation invariance The optimum of $\mathcal{J}_{\text{FIG}}(\tau)$ can change under a reparameterisation of θ: see Section 6 for a simple illustration. This property is undesirable as the optimal design is affected by the seemingly irrelevant choice of what parameterisation is used. In contrast, the design optimising expected Shannon information gain is reparameterisation invariant.

Intuitively uninformative designs By construction, optimal FIG designs maximise a particular definition of informativeness. However these designs often have properties which seem "uninformative" in an intuitive sense.

[3] Some preprints of this paper used this name for $\mathcal{U}_{\text{trace}}$ instead. Our usage here is a closer analogy to Shannon information gain, and matches that of Overstall (2022).

One issue is that FIG designs often result in posterior distributions which are very diffuse for some linear combinations of parameters, although they may be concentrated for others. This is illustrated in Figure 5, where the design results in a long narrow posterior in which the marginals for two parameters are very similar to the prior marginals. In contrast, a competing design produces posterior marginals which are concentrated around the true values for both parameters.

Also, FIG designs can have an excessive amount of *replication*: repeated observations at the same time or location. For example, supplementary Section I.5 shows a FIG design with *all* observations at the same time. Such repeated observations provide increasingly accurate inference on one parameter (or parameter combination) at the expense of the others.

An informal explanation for this behaviour is as follows. The optimal FIG design maximises the trace of $\bar{\mathcal{I}}(\tau)$, which equals the sum of its eigenvalues. Often this sum is maximised when one eigenvalue is large and the others are small: we provide an example in Section 6, and also Overstall (2022) describes a linear model where this occurs. In a Bayesian setting, $\mathcal{I}(\theta;\tau)$ is an approximation to posterior precision (see e.g. Van der Vaart, 2000), so this corresponds to the typical posterior having one parameter combination which is accurately inferred at the expense of the others. Overstall (2022) also points out that the optimal τ can produce a singular $\bar{\mathcal{I}}(\tau)$ matrix, causing some statistical methods to break down, and a reason for this is that $\mathcal{J}_{\text{FIG}}$ only uses terms on the diagonal of $\bar{\mathcal{I}}(\tau)$, allowing off-diagonal terms to make it singular.

4 Theory: game theory approach

The previous section provided a theoretical framework supporting the use of $\mathcal{U}_{\text{trace}}$, but found it produced designs with several undesirable features, including dependence on the choice of parameterisation and a diffuse marginal posterior for some linear combinations of parameters. Here we modify the framework to produce a design which is robust to linear reparameterisation, which, as we shall see, encourages marginal posterior concentration for all parameter combinations.

Below, Section 4.1 outlines our framework. Section 4.2 reviews some game theory definitions, which are used in Section 4.3 to characterise optimal designs supported by the framework, including an adversarial variation on FIG which we refer to as ADV. Section 4.4 describes the properties of ADV and its advantages over FIG.

4.1 Game theoretic framework

We propose the following game theoretic framework. Initially, the experimenter selects a design τ. We introduce a *critic* who now selects the parameterisation by choosing an invertible matrix A defining parameters $\phi = A^{-1}\theta$. The experimenter must then select a, a density for ϕ, receiving a reward $\mathcal{R}(a,\phi)$. The critic's reward is $-\mathcal{R}(a,\phi)$: they aim to find the parameterisation for which the design does worst. This completes the specification of a game, as discussed further in Section 4.2.

We continue to use assumption A1, which now gives a scoring rule of $\mathcal{S}(a, \phi) = -\mathcal{R}(a, \phi)$. We also introduce another assumption, which is discussed in supplementary Section D:

A3 The critic is restricted to selecting A with $\det A = 1$.

Table 2 in the supplementary material summarises and contrasts both the decision-theoretic and game-theoretic frameworks.

4.2 Game theory definitions

We refer to τ, A, θ, y, a as *actions* of our game. The game specifies a mapping from τ, A, θ, y, a to real-valued rewards for the *players*: the experimenter and critic. The actions θ, y are random samples from specified distributions. The remaining actions are chosen by the players according to *decision rules*, $\tau, A(\tau), a(\tau, A, y)$.

In the remainder of this section we explore the set of *subgame perfect equilibria* (SPEs): players act to optimise their expected reward under the assumption that later decisions will also do so. In our setting, SPE is a condition on decision rules. The first condition is that $a(\tau, A, y)$ must output some a maximising the experimenter's expected reward given (τ, A, y). The second condition is that $A(\tau)$ must output A maximising the critic's expected reward given τ and $a(\tau, A, y)$. Finally τ must maximise the experimenter's expected reward given $A(\tau)$ and $a(\tau, A, y)$. Note that existence and uniqueness of decision rules meeting these conditions is not automatic.

SPEs are often argued (see e.g. Jin et al., 2020) to be an appropriate solution concept for games in which players act sequentially. The alternative solution concept of *Nash equilibria* is more appropriate for games with simultaneous actions. For more background on game theory and solution concepts see e.g. Osborne and Rubinstein (1994).

4.3 Results

Our first result is that under logarithmic score, the game theoretic framework is essentially equivalent to the decision theoretic framework.

Result 3. *The following sets are equal:*

- *Designs which are selected in SPEs of the game theoretic framework with logarithmic score and assumptions A1, A3.*
- *Optimal designs in the decision theoretic framework with assumption A1 and logarithmic score.*

The next result shows that for Hyvärinen score, the game theoretic framework addresses the issue of sensitivity to reparameterisation which occurred for the decision theoretic framework. The result focuses on the case of a *linear reparameterisation* which we define as reparameterising θ to $B\theta$ for some invertible B.

Result 4. *The set of designs which are selected in SPEs of the game theoretic framework with Hyvärinen score and assumptions A1, A3 is unchanged by a linear reparameterisation.*

Finally, our main result characterises which designs are selected in SPEs under Hyvärinen score. It uses the following technical assumption:

A4 There exists some τ such that $\mathbb{E}_{\theta\sim\pi(\theta)}[\det \mathcal{I}(\theta;\tau)] > 0$.

Informally this means there is a design under which the model is guaranteed to provide some information on every parameter, or linear combination of parameters. We also use a regularity condition A5 (see supplementary Section A).

Result 5. *Consider the game theoretic framework with Hyvärinen score under assumptions A1, A3–A5. Then the set of τ, A pairs which are selected in SPEs are those which solve $\min_\tau \max_A \mathcal{K}(\tau, A)$ where*

$$\mathcal{K}(\tau, A) = -\,\mathbb{E}_{\theta\sim\pi(\theta)}\operatorname{tr}[A^T\mathcal{I}(\theta;\tau)A]. \tag{13}$$

Also, the set of designs which are selected in SPEs are those maximising

$$\mathcal{J}_{ADV}(\tau) = \det \bar{\mathcal{I}}(\tau), \tag{14}$$

where $\bar{\mathcal{I}}(\tau) = \mathbb{E}_{\theta\sim\pi(\theta)}[\mathcal{I}(\theta;\tau)]$.

4.4 Adversarial objective properties

Using the decision theoretic framework with Hyvärinen score produced the objective $\mathcal{J}_{\mathrm{FIG}}(\tau) = \operatorname{tr}\bar{\mathcal{I}}(\tau)$. Result 5 shows that the game theoretic framework instead produces the objective $\mathcal{J}_{\mathrm{ADV}}(\tau) = \det\bar{\mathcal{I}}(\tau)$. This has improved properties, as we discuss in this subsection.

Result 5 also shows that maximisation of $\mathcal{J}_{\mathrm{ADV}}(\tau)$ is equivalent to minimax optimisation of $\mathcal{K}(\tau, A)$, as defined in (13). This is helpful in discussing the properties of $\mathcal{J}_{\mathrm{ADV}}(\tau)$, and particularly useful in defining a practical optimisation scheme, as detailed below and in Section 5.

Linear reparameterisation invariance By Result 4, the set of optimal designs from $\mathcal{J}_{\mathrm{ADV}}(\tau)$ is invariant to linear reparameterisation, unlike $\mathcal{J}_{\mathrm{FIG}}(\tau)$. This is also easy to show directly. Consider a linear reparameterisation $\phi = B\theta$ with $\det B \neq 0$. From (6), the FIM is $\mathcal{I}_\phi(\phi;\tau) = B^{-T}\mathcal{I}_\theta(\theta;\tau)B^{-1}$. Thus $\bar{\mathcal{I}}_\phi(\tau) = B^{-T}\bar{\mathcal{I}}_\theta(\tau)B^{-1}$ and the reparameterised objective $\mathcal{J}_{\phi,\mathrm{ADV}}(\tau)$ equals $\mathcal{J}_{\theta,\mathrm{ADV}}(\tau)(\det B)^{-2}$: the original objective multiplied by a positive constant. Hence the set of optimal designs is unchanged.

Equivalent A matrices From the cyclic property of trace,

$$\mathcal{K}(\tau, A) = -\,\mathbb{E}_{\theta\sim\pi(\theta)}\operatorname{tr}[AA^T\mathcal{I}(\theta;\tau)].$$

Hence any two A matrices producing the same AA^T are equivalent in that they give the same $\mathcal{K}(\cdot, A)$ function. In particular, given any A, a Cholesky factor of AA^T is equivalent. So when we perform minimax optimisation, we can restrict A to be a Cholesky matrix (i.e. lower triangular with positive diagonal entries) with determinant 1. This will help set up our optimisation algorithm later – see Section 5.2.

Intuitively informative designs As discussed in Section 3.4, $\mathcal{J}_{\text{FIG}}(\tau)$ sometimes produces designs in which one parameter combination is inferred accurately but others are not. The objective $\mathcal{J}_{\text{ADV}}(\tau)$ penalises such designs. This is because it is the product of the eigenvalues of $\bar{\mathcal{I}}(\tau)$. Thus, compared to $\mathcal{J}_{\text{FIG}}(\tau)$, it is much less advantageous to make one eigenvalue large and the others small. See Figure 5 for an illustration – the posterior under the ADV design is concentrated with respect to both parameters shown in this plot, unlike that under the FIG design.

A related property of the objective $\mathcal{J}_{\text{FIG}}(\tau)$ is that it sometimes produces singular or near-singular $\bar{\mathcal{I}}(\tau)$ matrices. The objective $\mathcal{J}_{\text{ADV}}(\tau)$ avoids this by directly penalising matrices with low determinants. One reason this is possible is that the determinant operation is affected by off-diagonal elements of $\bar{\mathcal{I}}(\tau)$, unlike the trace.

Bayesian justification Unlike $\mathcal{J}_{\text{FIG}}(\tau)$, $\mathcal{J}_{\text{ADV}}(\tau)$ is not defined as the expectation of a utility function. Therefore the "pseudo-Bayesian" or "fully Bayesian" definitions of Ryan et al. (2016) discussed in Section 2.4 cannot be directly applied. However, $\mathcal{J}_{\text{ADV}}(\tau)$ meets the spirit of the pseudo-Bayesian definition as it depends on FIMs rather than posteriors. Nonetheless, we have shown it emerges from a game theoretic framework based on enabling an experimenter to estimate the posterior well.

Link to D-optimality Classical optimal design using D-optimality requires a design to maximise $\det \mathcal{I}(\theta; \tau)$ at a reference θ value. This can be generalised to a Bayesian setting incorporating parameter uncertainty in several ways. See Atkinson et al. (2007), Table 18.1, for a list of 5 possible objectives, including $\mathcal{J}_{\text{ADV}}(\tau)$ (and the objective of Pronzato and Walter (1985), mentioned in Section 1.2, which differs from $\mathcal{J}_{\text{ADV}}(\tau)$ by swapping the order of determinant and expectation.) One contribution of our work is to provide theoretical support and an optimisation method for this particular choice.

Computational tractability It is hard to optimise $\mathcal{J}_{\text{ADV}}(\tau)$ directly for continuous τ, as it is not easy to obtain an unbiased gradient estimate due to the non-linearity of the determinant operator. However optimisation based on (13) is tractable, as described in the next section.

5 Optimisation

Result 5 of the previous section motivates performing optimal design by finding minimax solutions of $\mathcal{K}(\tau, A) = -\mathbb{E}_{\theta \sim \pi(\theta)} \operatorname{tr}[A^T \mathcal{I}(\theta; \tau) A]$. Since this is an expectation, computation of unbiased gradient estimates is straightforward when $\mathcal{I}(\theta; \tau)$ can be evaluated.

(See supplementary Section G for discussion of the case where this is not possible.) This section describes how as a consequence it is possible to find candidate minimax solutions using generic gradient based optimisation methods. Throughout we assume $\nabla_\tau \mathcal{I}(\theta;\tau)$ and $\nabla_\tau \mathbb{E}_{\theta\sim\pi(\theta)} \mathcal{I}(\theta;\tau)$ exist.

Section 5.1 gives background on gradient based optimisation methods for minimax problems. Section 5.2 discusses how to deal with the constraint $\det A = 1$. Section 5.3 describes calculation of unbiased gradient estimates. Section 5.4 presents our optimisation algorithm, and describes various implementation details.

5.1 Gradient descent ascent

Algorithm 1, *gradient descent ascent*, attempts to solve $\min_x \max_y f(x,y)$. It iteratively updates (x_t, y_t) based on $g_{x,t}$ and $g_{y,t}$, unbiased gradient estimates of $-\nabla_x f(x_t, y_t)$ and $\nabla_y f(x_t, y_t)$. For an overview and history of GDA see Lin et al. (2020). GDA generalises *stochastic gradient descent* (SGD), which is the special case where f is a function of x alone, and only x_t updates are needed.

Algorithm 1 Gradient descent ascent (GDA)

```
Input: Initial values $x_1, y_1$, update subroutines $h_x, h_y$.
for $t = 1, 2, \dots$ do
    Compute $g_{x,t}, g_{y,t}$, unbiased estimates of $-\nabla_x f(x_t, y_t)$ and $\nabla_y f(x_t, y_t)$.
    Update estimates using $x_{t+1} = x_t + h_x(g_{x,t})$, $y_{t+1} = y_t + h_y(g_{y,t})$.
end for
```

A simple update rule for GDA, which we will refer to as the *default* update rule, is $z_{t+1} = z_t + \alpha_{z,t} g_{z,t}$ (for $z \in \{x, y\}$), given predefined learning rate sequences $\alpha_{x,t}, \alpha_{y,t}$. Stochastic approximation theory (see e.g. Kushner and Yin, 2003) suggests using learning rates such that $\sum_{t=1}^\infty \alpha_{z,t} = \infty$, $\sum_{t=1}^\infty \alpha_{z,t}^2 < \infty$ (for $z \in \{x, y\}$). This ensures that SGD using this update rule and unbiased gradient estimates converges to a local minimum, under appropriate regularity conditions.

More sophisticated update rules have been developed, effectively tuning the learning rates adaptively and using different learning rates for each component of the x and y vectors. We use the popular Adam update rule (Kingma and Ba, 2015). (We use the default Adam tuning parameters, which typically produce good empirical performance but not asymptotic convergence. Convergence guarantees are possible by setting some parameters to decay: see Kingma and Ba, 2015.)

Convergence of GDA under any update rule is more complicated than SGD, and is an area of active research. One issue is that the dynamics can produce limit cycles as well as limit points. Algorithms to avoid this have been suggested, including a *two timescale update rule* (Borkar, 1997; Heusel et al., 2017) in which $\lim_{t\to\infty} \alpha_{y,t}/\alpha_{x,t} = \infty$ (or a similar condition under the Adam learning rule.) Another issue is to characterise the limit points of GDA as an appropriate local generalisation of minimax solutions (Heusel et al., 2017; Jin et al., 2020; Lin et al., 2020).

However, for our experimental design application, we find empirically that standard GDA methods with the Adam update rule suffice to produce sensible designs. Therefore we recommend using GDA with this update rule, and checking for convergence using diagnostics, multiple runs from different initial values, and post-processing. Details of these are contained in the following subsections.

5.2 Representation of A

We wish to solve $\min_\tau \max_A \mathcal{K}(\tau, A)$ under the constraint that $\det A = 1$. As discussed in Section 4.4, it is sufficient to search for A over Cholesky matrices (i.e. lower triangular with positive diagonal entries) with determinant 1. Such matrices can be represented as

$$A(\eta) = \begin{pmatrix} \exp(\eta_{11}) & 0 & \dots & 0 \\ \eta_{21} & \exp(\eta_{22}) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{p1} & \eta_{p2} & \dots & \exp(-\sum_{i=1}^{p-1} \eta_{ii}) \end{pmatrix}. \tag{15}$$

This maps an unconstrained real vector η of η_{ij} variables to the space of matrices of interest. We can now solve $\min_\tau \max_\eta \mathcal{K}(\tau, A(\eta))$ using GDA. We initialise $\eta = 0$ so that initially $A = I$, corresponding to the critic making no reparameterisation.

5.3 Gradient estimation

In this section we will consider estimating $\nabla_z \mathcal{K}(\tau, A(\eta))$ for $z \in \{\tau, \eta\}$ in the common case where it is easy to evaluate the FIM and related gradients. See supplementary Section G for discussion of cases where these are intractable.

In a few cases it is possible to directly evaluate $\bar{\mathcal{I}}(\tau) = \mathbb{E}_{\theta \sim \pi(\theta)}[\mathcal{I}(\theta; \tau)]$. See Sections 6 and 8 for example. Then, using (13), we have $\mathcal{K}(\tau, A) = -\operatorname{tr}[A^T \bar{\mathcal{I}}(\tau) A]$, and gradients can be evaluated using automatic differentiation (Baydin et al., 2017) or derivation of direct expressions for them. Our code performs automatic differentiation using the PyTorch library (Paszke et al., 2019).

More commonly it is necessary to derive unbiased gradient estimates. From the definition of $\mathcal{K}(\tau, A)$, (13), an unbiased estimate is

$$\hat{\mathcal{K}}(\tau, A) = -\operatorname{tr}\left[A^T \left\{ \frac{1}{K} \sum_{k=1}^{K} \mathcal{I}(\theta^{(k)}; \tau) \right\} A \right]. \tag{16}$$

We also have the following result.

Result 6. *Under appropriate regularity conditions on* $\operatorname{tr}[A^T \mathcal{I}(\theta; \tau) A]$ *(see supplementary Section F), then* $\nabla_z \hat{\mathcal{K}}(\tau, A(\eta))$ *is an unbiased estimate of* $\nabla_z \mathcal{K}(\tau, A(\eta))$ *for* $z \in \{\tau, \eta\}$.

We calculate gradients of $\hat{\mathcal{K}}(\tau, A)$ using automatic differentiation in PyTorch. Note that using a larger K in (16) reduces the variance but increases computational cost. We explore this trade-off in Section 7.

5.4 Main algorithm

Algorithm 2 applies GDA to our experimental design setting. Note that we perform R replications of GDA in parallel from different initial conditions. This is typically more efficient than repeating the algorithm R times in serial. One reason is that the calculations in step 3 and 4 are amenable to parallelisation. (We ran our experiments on a CPU, but our PyTorch code can easily exploit GPU parallelisation, allowing for further speed improvements.) Another reason is that the same simulations in step 3 can be reused for all replications. Algorithm 2 can also be used for SGD to optimise the FIG objective by keeping $\eta = 0$ and only updating τ.

Algorithm 2 Gradient descent ascent for Bayesian experimental design

```
1: Input: Number of samples to use in (16) $K$, number of parallel replications to
   perform $R$, initial values $\tau_1^i, \eta_1^i$ (for $i = 1, 2, \dots, R$), update subroutines $h_\tau, h_\eta$.
2: for $t = 1, 2, \dots$ do
3:     Sample $\theta^{(k)}$ from the prior for $k = 1, 2, \dots, K$.
4:     Compute $g_{\tau,t}^i, g_{\eta,t}^i$, unbiased estimates of $-\nabla_\tau \mathcal{K}(\tau_t^i, A(\eta_t^i))$ and $\nabla_\eta \mathcal{K}(\tau_t^i, A(\eta_t^i))$
       using automatic differentiation[4]of (16) for all $i$.
5:     Update estimates using $\tau_{t+1}^i = \tau_t^i + h_\tau(g_{\tau,t}^i)$, $\eta_{t+1}^i = \eta_t^i + h_\eta(g_{\eta,t}^i)$ for all $i$.
6: end for
```

In the remainder of this section various implementation details are discussed.

Diagnostics

Step 4 of Algorithm 2 involves calculating $\hat{\mathcal{K}}(\tau, A)$ from (16), whose gradients are then found using automatic differentiation. This Monte Carlo estimate of $\mathcal{K}$ can be used as a diagnostic of the algorithm's performance. As we are performing minimax optimisation, it will typically rise and fall before reaching an equilibrium. Also the values of $\hat{\mathcal{K}}$ from different parallel runs can be highly correlated since they are based on the same $\theta^{(k)}$ samples. Both phenomena can be seen in the bottom left graph of Figure 2. (The presence of correlation is illustrated by the fact that the right hand side of the graph seems to show a single thick line. In fact there are multiple lines with different values which are highly correlated with each other.)

An alternative diagnostic is to estimate $\mathcal{J}_{\text{ADV}}$ as defined in (14) by

$$\hat{\mathcal{J}}_{\text{ADV}}(\tau) = \det\left(\frac{1}{J}\sum_{j=1}^{J} \mathcal{I}(\tilde{\theta}^{(j)}; \tau)\right). \tag{17}$$

Our code can optionally calculate $\hat{\mathcal{J}}_{\text{ADV}}$ for designs produced during its execution. To do so we initially sample $\tilde{\theta}^{(j)}$ values from the prior for $j = 1, 2, \dots, J$ (we take $J = 1000$). Then $\hat{\mathcal{J}}_{\text{ADV}}$ is calculated for each design τ_t^i produced during the algorithm. Unlike $\hat{\mathcal{K}}$,

[4] In practice we implement this by differentiating $\sum_{i=1}^{R} \hat{\mathcal{K}}(\tau_t^i, A(\eta_t^i))$ with respect to all τ, η variables. This is more efficient in backwards mode automatic differentiation, as implemented in PyTorch.

this adds an extra computational cost. However this diagnostic has the advantage that it directly estimates the objective $\mathcal{J}_{\text{ADV}}$ so larger values correspond to better performing designs. For example, the bottom right graph of Figure 2 directly traces the performance of designs during optimisation.

For SGD optimisation we can return $\hat{\mathcal{K}}(\tau, I)$ as a diagnostic. This directly estimates $-\mathcal{J}_{\text{FIG}}(\tau)$ but does so using a different θ sample each time, which adds some variability to the diagnostic. To remedy this we can also calculate

$$\hat{\mathcal{J}}_{\text{FIG}}(\tau) = \operatorname{tr}\left(\frac{1}{J}\sum_{j=1}^{J}\mathcal{I}(\tilde{\theta}^{(j)};\tau)\right)$$

for a fixed sample of $\tilde{\theta}^{(j)}$ values, as above.

Termination

We run Algorithm 2 for a fixed number of iterations or fixed runtime. Alternatively it could be run until a convergence condition is met for one of the diagnostics above, or for the size of the increments to τ or A.

Optimisation under constraints

We often wish to find the optimal design under a constraint: $\tau \in \mathcal{T} \subset \mathbb{R}^d$. In our applications we achieved this using simple pragmatic approaches. In Section 6 we represent (scalar) τ as the transformation of an unconstrained variable. For most examples in Section 7 the designs remain in $\mathcal{T}$ under unconstrained optimisation, so no modification to this is needed. We address constraints for analyses that require a minimum time between observations in Section 7 or in Section 8 by adding a large penalty to $\mathcal{K}$ for $\tau \notin \mathcal{T}$, whose gradient moves designs towards $\mathcal{T}$. In more complex settings these methods may not suffice. A more sophisticated alternative would be to compose each GDA update with a projection operation into $\mathcal{T}$ (Kushner and Yin, 2003).

Local optima

GDA can often converge to multiple possible locally optimal designs. To attempt to find the global optimum we run GDA multiple times from different initial values of τ (keeping η initialised as a zero vector), and compare their $\hat{\mathcal{J}}$ diagnostics.

In some settings we can also use a *point exchange algorithm*. Suppose we must select multiple *design points* from some region. Optimal designs often have a high degree of *replication*: they consist of a small number of clusters of repeated observations (Gotwalt et al., 2009; Overstall and Woods, 2017; Binois et al., 2019). While gradient based optimisation can find the optimal cluster locations well, there may be a large number of local optima, differing by the number of points in each cluster. See Figure 3 for an example. A point exchange algorithm takes cluster locations as input and uses discrete optimisation to find optimal cluster sizes. We use a simple approach detailed in supplementary Section E. There is scope for developing more sophisticated approaches in future work.

6 Poisson model illustration

This section provides a simple illustrative example of the properties of the SIG, FIG and ADV approaches to experimental design. The setting is that an experimenter must divide a unit of time between two experiments making Poisson observations with different rates. More precisely, the design is $\tau \in [0, 1]$. There are 2 independent observations $y_1 \sim Poisson(\tau\theta_1\omega_1), y_2 \sim Poisson([1-\tau]\theta_2\omega_2)$. We assume $\omega_1 > \omega_2 > 0$ and that θ_1, θ_2 have independent $Gamma(2, 1)$ priors.

Analytic results Optimal designs for this example can be derived analytically. We summarise the results here: see supplementary Section H for derivations. The optimal design is $\tau = 1$ under FIG and $\tau = 1/2$ under SIG or ADV. Under the FIG design, y_2 is always zero, so this design produces no information on θ_2 i.e. the posterior always equals the prior. The SIG/ADV design avoids this undesirable property. Also, the SIG/ADV design is invariant to linear reparameterisations. However for FIG, linear reparameterisation can change the optimal design to $\tau_1 = 0$, or make all values of τ optimal.

Numerical optimisation We perform a numerical analysis of this example with $\omega_1 = 2, \omega_2 = 1$. Since the design $\tau \in [0, 1]$ is bounded we optimise a transformation, $\lambda = \text{logit}(\tau)$, to allow the use of unconstrained optimisation. Following (15) we take the critic's action to be

$$A(\eta) = \begin{pmatrix} \exp(\eta_{11}) & 0 \\ \eta_{21} & \exp(-\eta_{11}) \end{pmatrix}.$$

Figure 1 shows a GDA vector field using default update rules with a particular choice of learning rates. The figure shows λ and η_{11} for η_{21} fixed at zero. (In practice other η_{21} values quickly converge to zero.) The vector field illustrates spiral trajectories converging to a limit point. Additionally, it shows that for any fixed value of η_{11}, $\frac{\partial \mathcal{K}}{\partial \lambda}$ has a fixed sign. This illustrates that for fixed A (i.e. the FIG setting), SGD optimisation produces a design converging to either $\tau = 0$ or 1.

The figure also shows GDA trajectories as the critic learning rate is varied. In all cases, the trajectories converge on the limit point. However, convergence is much faster as the critic's learning rate is increased.

7 Pharmacokinetic example

This section contains simulations studies on a pharmacokinetic model. The main goal is to investigate the performance of our FIG and ADV approaches and compare them with existing methods for SIG.

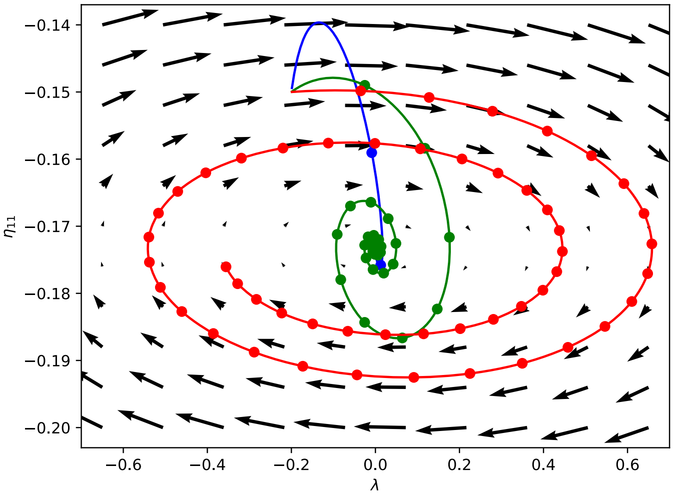

Figure 1: GDA vector field and trajectories for the Poisson example. Paths are shown for experimenter learning rate 10^{-2} and critic learning rate 10^{-3} (blue), 10^{-4} (green), 10^{-5} (red). Points indicate every 500 steps of optimisation. The vector field corresponds to the 10^{-5} critic learning rate.

7.1 Model

Pharmacokinetics studies the time course of drug absorption, distribution, metabolism, and excretion. Concentration level is observed via samples of fluid – such as blood, plasma or urine – from the subject taken at preplanned time points. The number of observations is constrained by budget and resources, as well as patient comfort and wellbeing. We assume observed concentration, y_i, at time τ_i (in hours) is distributed as

$$y_i \sim N(x(\theta, \tau_i), \sigma^2), \qquad \text{where } x(\theta, \tau_i) = \frac{D\theta_2(\exp[-\theta_1\tau_i] - \exp[-\theta_2\tau_i])}{\theta_3(\theta_2 - \theta_1)},$$

and $D = 400$. Concentrations at different times are assumed to be independent. We assume independent log normal priors

$$\theta_1 \sim LN(\log 0.1, 0.05), \theta_2 \sim LN(\log 1, 0.05), \theta_3 \sim LN(\log 20, 0.05),$$

and aim to find 15 observation times in $[0, 24]$. Also we treat $\sigma^2 = 0.1$ as known.

A similar model to the above was used by Ryan et al. (2014) and Overstall and Woods (2017). However we make two modifications to create a simple setting for comparing different methods. First, we omit a multiplicative noise term. Secondly, we do not enforce a 15 minute minimum time gap between consecutive observations, as its implementation would vary between methods making it more difficult to draw fair conclusions. In Section 7.5 we remove these modifications and show results from our approach for the realistic model used in previous work.

7.2 Methods

The FIM for this model is available in closed form – see supplementary Section I.1. This allows the optimal ADV design to be found using GDA (Algorithm 2), and the optimal FIG design using a SGD variant (only updating τ). In both cases we used the Adam update rule. When implementing these algorithms, we found no need to use constrained optimisation as the designs remained in the interval $[0, 24]$ in any case. We performed 100 replications in parallel with each algorithm, based on 100 initial designs sampled from a uniform distribution over $[0, 24]^{15}$. We considered several choices of K, the number samples to estimate $\mathcal{K}$ in (16), and selected $K = 1$. See supplementary Section I.2 for details.

We compare our results to two methods of finding the optimal SIG design: the approximate coordinate exchange (ACE) algorithm of Overstall and Woods (2017), and the prior contrastive estimation (PCE) algorithm of Foster et al. (2020). (PCE is one of several methods in Foster et al. 2020, and is not their overall recommendation. However we found PCE converged more quickly than the alternatives for this example, so we use it as lower bound on the speed of their methods.) Implementation details of these methods are given in Appendices I.3 and I.4. PCE allowed 100 replications to be run in parallel. We ran replications of ACE serially, noting that the ACE code utilises multiple cores during its execution in any case. As ACE took much longer to run we used only 30 replications.

7.3 Results

Table 1 shows mean times for each method i.e. run time divided by number of designs returned, demonstrating the speed advantage of GDA and SGD. Below we comment in more detail on each algorithm's results.

	GDA	GDA+PE	SGD	SGD+PE	ACE	PCE
Mean time (seconds)	1.4	2.2	1.4	1.5	8012	36
Number of repetitions	100	100	100	100	30	100

Table 1: Mean times to run optimisation methods on the pharmacokinetic example. (n.b. PE is "point exchange"). Both ACE and PCE have not fully converged after the time quoted, so the figures given are lower bounds on the run time.

ADV The top half of Figure 2 shows trajectories of τ and A during a single replication of GDA optimisation. The design points eventually settle into clusters of repeated observations at three times. The bottom half shows estimated $\mathcal{K}$ and $\mathcal{J}$ objectives over 100 replications. Although all runs have converged after 100,000 iterations, the objective values are not identical.

Figure 3 shows final designs for all replications. The design points typically converge to 3 clusters, around times 1.1, 3.4 and 14.0. However the cluster sizes vary between runs, as different local optima are found. This explains why runs converge to slightly different objective values. Employing point exchange reduces the variation in cluster

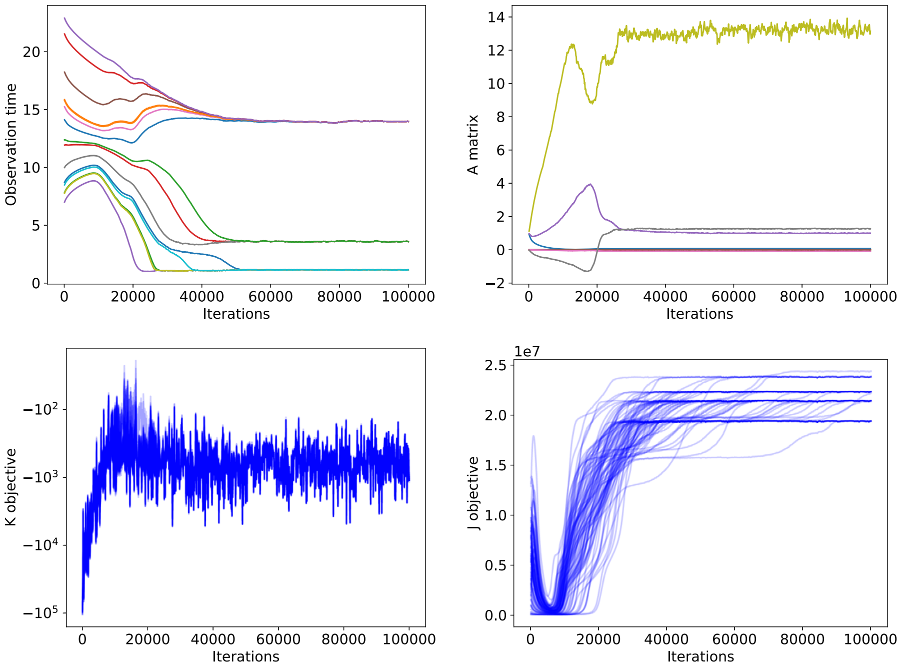

Figure 2: Trace plots for GDA on the pharmacokinetic example. The top row shows output for a single run of GDA. The bottom row shows output for 100 runs.

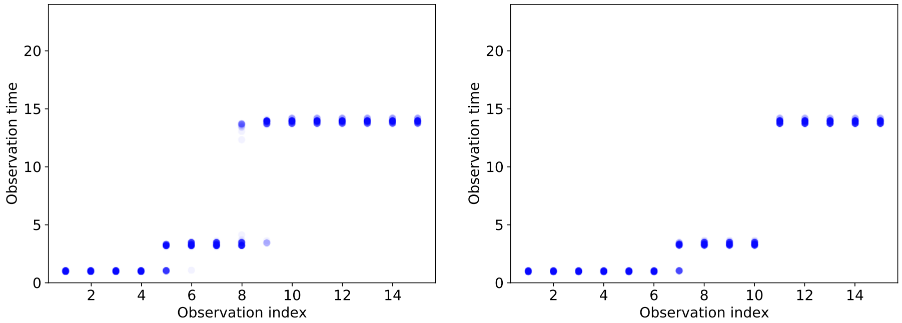

Figure 3: Designs output from 100 optimisation runs for the pharmacokinetic example. The horizontal axis shows the index of each point in the sorted design i.e. observation times are shown in increasing order from left to right. Points are plotted as semi-transparent, so rare results are light, and common results are dark. The left plot shows GDA output after 100,000 iterations. In the right plot point exchange is also applied.

sizes. (There is some variability in exact cluster locations. Further investigation showed that this is mainly due to cluster locations changing slightly depending on cluster sizes.) PE finds two candidates for an optimal design, with 6 or 7 points near time 1. (To find

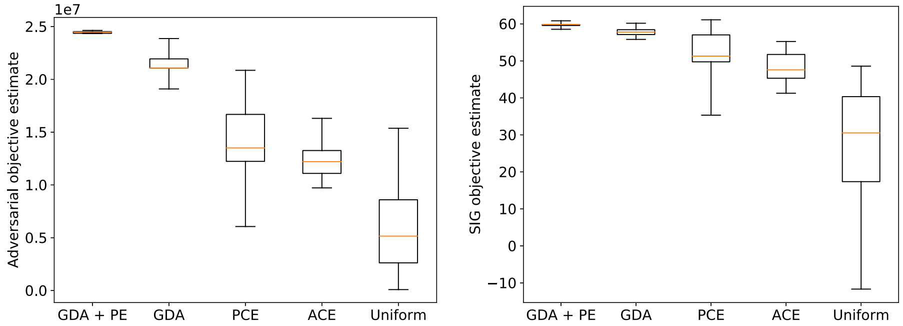

Figure 4: Performance of pharmacokinetic example designs output by various methods. Performance is judged by estimated objective value achieved for (left) our adversarial approach and (right) the Shannon information gain approach. Boxplots are shown for estimates taken from 100 designs (or 30 in the case of ACE) from repeated optimisation runs with different initialisations.

an overall optimum one could now estimate $\hat{\mathcal{J}}_{\mathrm{ADV}}$ for both using a larger sample size.) However GDA runs produced a maximum of 6 points for this cluster. This highlights the importance of PE to find global optima.

Figure 4 (left) shows $\hat{\mathcal{J}}_{\mathrm{ADV}}$ values achieved by the final designs from GDA and the SIG methods, as well as for designs sampled from a uniform distribution over $[0, 24]^{15}$. These estimated were calculated using equation (17). On average GDA outperforms the SIG methods and further improvement is achieved by also applying point exchange.

Finally, Figure 5 illustrates a posterior found from one particular ADV design, and shows that it is concentrated around the true parameters.

FIG Full details of SGD optimisation results are given in supplementary Section I.5. Briefly, SGD designs always converge to a single cluster around time 12. Intuitively this is a poor design: the three θ parameters of the $x(\theta, \tau)$ function cannot all be identified from repeated observations at a single time point τ. Figure 5 illustrates that this design indeed performs poorly by presenting a typical posterior. The narrowness of the FIG design posterior illustrates that the posterior is highly concentrated for some function of θ_1 and θ_3. However the posterior is diffuse for θ_1 and θ_3 marginally: it stretches almost the full length of the prior distribution, unlike the highly concentrated ADV design posterior.

SIG Running ACE under the default tuning choices took a few hours, much slower than GDA which took only a few minutes to run 100 repetitions. Furthermore, the ACE results do not seem to have converged to the optimal design in the time. See supplementary Section I.3 for more details. (We also explored varying one tuning choice in ACE: the number of Monte Carlo samples for utility estimates used to fit the Gaussian process. However we found only marginal improvements over the recommended default tuning for ACE.)

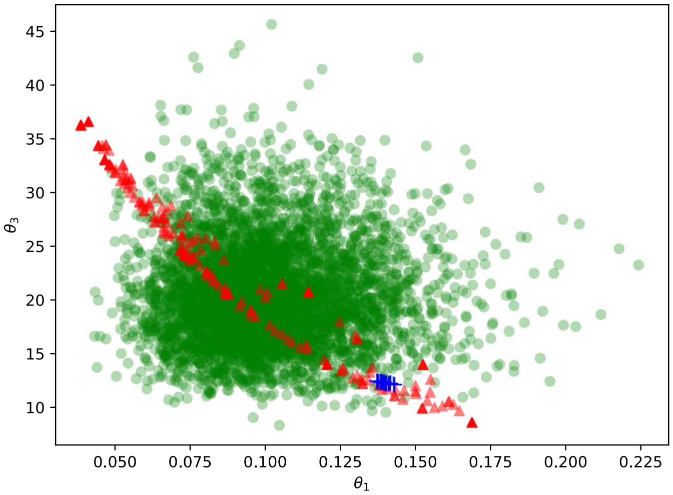

Figure 5: Bivariate posteriors for θ_1, θ_3 in the pharmacokinetic example. Samples are plotted from: the prior (green circles), the posterior with a FIG design (red triangles), the posterior with a ADV design (blue pluses). The ADV and FIG designs used 100,000 iterations of GDA or SGD, followed by point exchange optimisation. The true parameter values, sampled from the prior, are $\theta_1 = 0.14, \theta_3 = 12.29$, consistent with the blue pluses. Datasets for use in inference were then sampled from the model, using the same observation noise realisations for both ADV and FIG designs. Posterior samples are importance sampling output.

We ran PCE for roughly 10 times longer than GDA and found the results concentrated near 3 observation times similar to those for GDA. However this runtime appears to only be a lower bound on the time needed for convergence, as running PCE for longer increased the concentration of the design points. We also explored alternative methods from Foster et al. (2020) but were not able to improve on the PCE results. See supplementary Section I.4 for more details.

Figure 4 (right) shows estimates of expected Shannon information gain achieved by the final designs from all SIG methods, as well as GDA designs and designs sampled from a uniform distribution over $[0, 24]^{15}$. The calculation method is described in supplementary Section I.6. On average GDA outperforms the other methods, with a slight further improvement from using point exchange. This suggests that GDA designs produce good performance under the SIG objective, and is also further evidence that the ACE and PCE methods have not fully converged to the overall optimum.

7.4 Conclusion of comparisons

We have shown that our optimisation method to find ADV and FIG designs is faster than SIG optimisation methods by at least a factor of 10. The true advantage may be greater as the SIG methods do not appear to have fully converged in the time stated.

The FIG design can give overly diffuse marginal posteriors for some parameters. The ADV design avoids this drawback, and appears to be similar to the SIG design, and indeed gives competitive performance under the SIG objective. Multiple runs of ADV produced designs with similar cluster locations but varying cluster sizes. Post-processing these designs using point exchange further improved the ADV objective reached, illustrating the importance of this step. To explain the improvement, note that point exchange found two candidates for optimal cluster sizes, including one which was different from any cluster sizes found without post-processing.

7.5 Realistic example

Here we implement our ADV approach on a more realistic version of the pharmacokinetic example. Firstly, we now assume *multiplicative noise*:

$$y_i \sim N(x(\theta, \tau_i), \sigma_1^2 + \sigma_2^2 x(\theta, \tau_i)^2), \tag{18}$$

with $\sigma_1^2 = 0.1, \sigma_2^2 = 0.01$. Secondly we require *gaps* i.e. a constraint that observation times are at least 0.25 hours apart. These changes result in the model used by Ryan et al. (2014) and Overstall and Woods (2017).

Results exist for the FIM of a model with multiplicative noise (e.g. see Malagò and Pistone, 2015, equation 23). However a lengthy analytic derivation is required which can easily result in errors. Therefore we implement a method which can be used when the FIM is intractable: Algorithm 1, described in supplementary Section G. Supplementary Section J gives further implementation details for this example and more comments on the results.

Figure 6 shows the resulting designs. Without the gaps constraint, there are 3 clusters of repeated design points. The cluster locations differ depending on whether or not multiplicative noise is used. When gaps are enforced, the clusters remain at the same locations, but consist of spaced out, rather than repeated, design points.

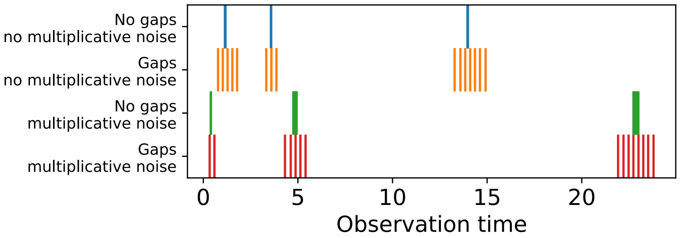

Figure 6: Optimal ADV designs for several variations of the pharmacokinetic example.

8 Geostatistical regression example

This section considers an example requiring hundreds of design choices, to illustrate how our method can scale up to higher dimensional applications.

8.1 Model

Consider the following geostatistical regression model. Here a design τ is a $d \times 2$ matrix whose rows specify measurement locations. We assume normal observations with a linear trend and squared exponential covariance function with a nugget effect, giving

$$y \sim N(x(\theta, \tau), \Sigma(\tau)), \qquad x_i = \theta_1 \tau_{i1} + \theta_2 \tau_{i2},$$

$$\Sigma = \sigma_1^2 I + \sigma_2^2 R(\tau), \qquad R_{ij} = \exp\left[-\sum_{k=1}^{2} (\tau_{ik} - \tau_{jk})^2 / \ell^2 \right].$$

For simplicity we assume that σ_1^2, σ_2^2 (observation variance components) and ℓ (covariance length scale) are known. Hence the unknown parameters are θ_1 and θ_2 (trends).

8.2 Methods

Using (7) the FIM is $\mathcal{I}(\tau) = \tau^T \Sigma(\tau)^{-1} \tau$, which does not depend on θ. Hence $\mathcal{I}(\tau)$ is also the expected FIM and we do not need to use Monte Carlo to estimate it.

We performed simulation studies for $\ell = 0.01, 0.02, 0.04, 0.08$ with $\sigma_1 = 1$ and $\sigma_2 = 3$ to search for 500 design points restricted to a unit square centred at the origin. The design was initialised as independent uniform draws. Each run used Algorithm 2 with the Adam update rule for 1000 iterations, which was enough for convergence (see supplementary Section K). We implemented constrained optimisation by adding a L_1 penalty to designs outside the unit square.

As this example aims to illustrate the time required by ADV, we did not investigate repeated runs from different initial designs, or post-processing using point exchange. In any case, the latter seems unlikely to help as there is little evidence of replicated observations in the results.

8.3 Results

Optimisation took on average 19.4 seconds. Figure 7 shows the resulting designs. For small ℓ values, the design points cluster in the corners. For larger values, the designs are spread across the region with varying spatial structures. For all runs A remained very close to the identity matrix throughout optimisation, reflecting the symmetry of θ_1 and θ_2 in the model. Hence FIG would also work well for this example.

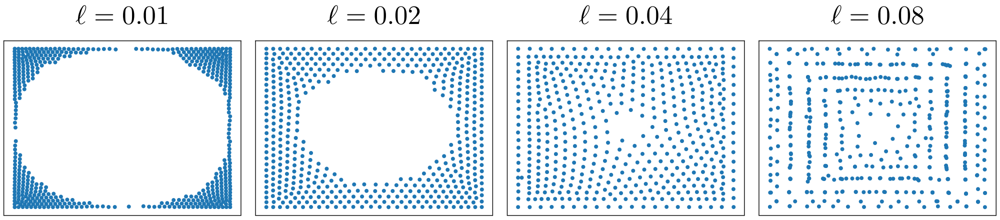

Figure 7: Geostatistical regression model designs returned for various choices of ℓ. The design space is a unit square centred at the origin.

9 Discussion

We have presented a gradient descent ascent algorithm Bayesian optimal design using an objective function based on the Fisher information. This provides improvements in speed and scalability to higher dimensional designs by avoiding the need for posterior inference. We also provide a novel game theoretic justification for our objective, and provide theoretical insights into the choice of utilities for Bayesian optimal design from decision/game theoretic principles, extending the work of Bernardo (1979) and Walker (2016).

In simulation studies our approach finds locally optimal design faster than other state of the art methods, by a factor of at least 10. To assess whether multiple locally optimal designs exist we recommend starting from multiple random initial designs, which can easily be done in a parallel version of our algorithm. If locally optimal designs involve clusters of repeated observations, we recommend post-processing using a point exchange algorithm. Although we did not observe any in our examples, GDA dynamics can converge to limit cycles rather than the desired solutions. We recommend checking for cyclic behaviour using trace plots of τ values produced during optimisation.

9.1 Limitations and future research

Intractable Fisher information Our methods have assumed that the FIM, and associated gradients, can easily be evaluated. When the FIM cannot be evaluated but the likelihood or score function (5) can, it is possible to produce an unbiased estimate of the FIM. Supplementary Section G describes this and how it can be used for experimental design. Section 7.5 contained an application.

However sometimes the likelihood and score function cannot be evaluated. One common reason is the presence of latent variables, such as nuisance parameters. Supplementary Section G describes difficulties of implementing our approach in this setting, and sketches methods to do so, which we plan to investigate in future research.

Discrete designs Gradient based optimisation is only available for a continuous space of possible designs. It would be of interest to develop analogous methods to this paper for discrete designs. These could be based on discrete optimisation algorithms, or involve relaxation of the discrete problem to a continuous approximation.

Discrete parameters A limitation of our work is that it does not apply to discrete parameters, since Hyvärinen score and FIM are only defined for continuous parameters. Discrete analogues have been proposed (Dawid et al., 2012; Shao et al., 2019), which would be interesting to investigate.

Variance reduction Optimisation efficiency could be increased by reducing the variance of our Monte Carlo gradient estimates. For instance, a reviewer suggests the use of randomised quasi-Monte Carlo, as in Drovandi and Tran (2018).

Alternative optimisers Variations on GDA with better convergence guarantees, such as two time scaled update rules (Heusel et al., 2017), are an active topic of research and could be used for the objective in this paper. Another possibility for future work is modifying generic gradient based optimisation methods to avoid local optima in optimal design problems e.g. using tempering methods, or non-local updates such as line search, as used by Overstall and Woods (2017).

Variations to game theoretic framework Several details of our game theoretic framework could be altered. One possibility is to consider alternative scoring rules. Many alternative proper scoring rules exist beyond logarithmic and Hyvärinen (Parry et al., 2012). Alternatively, non-proper scoring rules could be used which emphasise a particularly important aspect of inference to the task at hand e.g. the tails of the distribution (Loaiza-Maya et al., 2021). Another variation is to allow more freedom to the critic. This could include the ability to make more general, non-linear, reparameterisations, or to condition their reparameterisation on the observations, y.

Supplementary Material

Bayesian experimental design without posterior calculations: an adversarial approach. Supplementary material (DOI: 10.1214/22-BA1306SUPP; .pdf).

References

Amzal, B., Bois, F. Y., Parent, E., and Robert, C. P. (2006). Bayesian-optimal design via interacting particle systems. *Journal of the American Statistical Association*, 101(474):773–785. MR2281248. doi: https://doi.org/10.1198/016214505000001159. 133, 136

Atkinson, A., Donev, A., and Tobias, R. (2007). *Optimum experimental designs, with SAS*. Oxford University Press. MR2323647. 138, 145

Bandiera, L., Hou, Z., Kothamachu, V., Balsa-Canto, E., Swain, P., and Menolascina, F. (2018). On-line optimal input design increases the efficiency and accuracy of the modelling of an inducible synthetic promoter. *Processes*, 6(9):148. doi: https://doi.org/10.3390/pr6090148. 133

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2017). Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):5595–5637. MR3800512. 134, 147

Bernardo, J. M. (1979). Expected information as expected utility. *The Annals of Statistics*, 7(3):686–690. MR0527503. doi: https://doi.org/10.1214/aos/1176344689. 134, 135, 138, 139, 140, 158

Binois, M., Huang, J., Gramacy, R. B., and Ludkovski, M. (2019). Replication or exploration? Sequential design for stochastic simulation experiments. *Technometrics*, 61(1):7–23. MR3933655. doi: https://doi.org/10.1080/00401706.2018.1469433. 149

Borkar, V. S. (1997). Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291–294. MR1432654. doi: https://doi.org/10.1016/S0167-6911(97)90015-3. 146

Chaloner, K. and Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304. MR1390519. 138, 139

Cook, A. R., Gibson, G. J., and Gilligan, C. A. (2008). Optimal observation times in experimental epidemic processes. *Biometrics*, 64(3):860–868. MR2526637. doi: https://doi.org/10.1111/j.1541-0420.2007.00931.x. 133

Dawid, A. P., Lauritzen, S., and Parry, M. (2012). Proper local scoring rules on discrete sample spaces. *The Annals of Statistics*, 40(1):593–608. MR3014318. doi: https://doi.org/10.1214/12-AOS972. 158

Drovandi, C. C. and Tran, M.-N. (2018). Improving the efficiency of fully Bayesian optimal design of experiments using randomised quasi-Monte Carlo. *Bayesian Analysis*, 13(1):139–162. MR3737946. doi: https://doi.org/10.1214/16-BA1045. 158

Foster, A., Jankowiak, M., Bingham, E., Horsfall, P., Teh, Y. W., Rainforth, T., and Goodman, N. (2019). Variational Bayesian optimal experimental design. In *Advances in Neural Information Processing Systems*. 134, 136

Foster, A., Jankowiak, M., O'Meara, M., Teh, Y. W., and Rainforth, T. (2020). A unified stochastic gradient approach to designing Bayesian-optimal experiments. In *International Conference on Artificial Intelligence and Statistics*. 134, 136, 152, 155

Gillespie, C. S. and Boys, R. J. (2019). Efficient construction of Bayes optimal designs for stochastic process models. *Statistics and Computing*, 29(4):697–706. MR3955280. doi: https://doi.org/10.1007/s11222-018-9833-3. 133, 136

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378. MR2345548. doi: https://doi.org/10.1198/016214506000001437. 139

Gotwalt, C. M., Jones, B. A., and Steinberg, D. M. (2009). Fast computation of designs robust to parameter uncertainty for nonlinear settings. *Technometrics*, 51(1):88–95. MR2664866. doi: https://doi.org/10.1198/TECH.2009.0009. 149

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*. 134, 146, 159

Huan, X. and Marzouk, Y. M. (2013). Simulation-based optimal Bayesian experimental design for nonlinear systems. *Journal of Computational Physics*, 232(1):288–317. MR2994301. doi: https://doi.org/10.1016/j.jcp.2012.08.013. 136

Huan, X. and Marzouk, Y. M. (2014). Gradient-based stochastic optimization methods in Bayesian experimental design. *International Journal for Uncertainty Quantification*, 4(6). MR3275924. doi: https://doi.org/10.1615/Int.J.UncertaintyQuantification.2014006730. 136

Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709. MR2249836. 134, 140

Jin, C., Netrapalli, P., and Jordan, M. (2020). What is local optimality in nonconvex-nonconcave minimax optimization? In *International Conference on Machine Learning*. 134, 143, 146

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations*. 134, 146

Kleinegesse, S. and Gutmann, M. U. (2020). Bayesian experimental design for implicit models by mutual information neural estimation. In *International Conference on Machine Learning*. MR4303868. doi: https://doi.org/10.1214/20-BA1225. 136

Krause, A., Rajagopal, R., Gupta, A., and Guestrin, C. (2009). Simultaneous placement and scheduling of sensors. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 181–192. IEEE Computer Society. 133

Kück, H., de Freitas, N., and Doucet, A. (2006). SMC samplers for Bayesian optimal nonlinear design. In *2006 IEEE Nonlinear Statistical Signal Processing Workshop*, pages 99–102. IEEE. doi: https://doi.org/10.1109/NSSPW.2006.4378829. 136

Kushner, H. and Yin, G. G. (2003). *Stochastic approximation and recursive algorithms and applications*. Springer Science & Business Media. MR1993642. 146, 149

Lehmann, E. L. and Casella, G. (2006). *Theory of point estimation*. Springer Science & Business Media. MR1639875. 137

Lin, T., Jin, C., and Jordan, M. (2020). On gradient descent ascent for nonconvex-concave minimax problems. In *International Conference on Machine Learning*. 146

Lindley, D. V. (1956). On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005. MR0083936. doi: https://doi.org/10.1214/aoms/1177728069. 138

Loaiza-Maya, R., Martin, G. M., and Frazier, D. T. (2021). Focused Bayesian prediction. *Journal of Applied Econometrics*, 36(5):517–543. MR4309597. doi: https://doi.org/10.1002/jae.2810. 159

Malagò, L. and Pistone, G. (2015). Information geometry of the Gaussian distribution in view of stochastic optimization. In *ACM Conference on Foundations of Genetic Algorithms*, pages 150–162. MR3388148. doi: https://doi.org/10.1145/2725494.2725510. 156

Müller, P. (1999). Simulation-based optimal design. In *Bayesian Statistics 6: Proceedings of Sixth Valencia International Meeting*, pages 459–474. Oxford University Press. MR1723509. 134, 136

Oates, C. J., Cockayne, J., Prangle, D., Sullivan, T. J., and Girolami, M. (2020). Optimality criteria for probabilistic numerical methods. In Hickernell, F. J. and Kritzer, P., editors, *Multivariate Algorithms and Information-Based Complexity*. De Gruyter. MR4027836. doi: https://doi.org/10.1137/17M1139357. 133

Osborne, M. J. and Rubinstein, A. (1994). *A course in game theory*. MIT press. MR1301776. 143

Overstall, A. M. (2022). Properties of Fisher information gain for Bayesian design of experiments. *Journal of Statistical Planning and Inference*. MR4341631. doi: https://doi.org/10.1016/j.jspi.2021.10.006. 135, 139, 141, 142

Overstall, A. M. and Woods, D. C. (2017). Bayesian design of experiments using approximate coordinate exchange. *Technometrics*, 59(4):458–470. MR3740963. doi: https://doi.org/10.1080/00401706.2016.1251495. 133, 134, 136, 138, 149, 151, 152, 156, 159

Parry, M., Dawid, A. P., and Lauritzen, S. (2012). Proper local scoring rules. *The Annals of Statistics*, 40(1):561–592. MR3014317. doi: https://doi.org/10.1214/12-AOS971. 139, 159

Paszke, A. et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*. 147

Prangle, D., Harbisher, S., and Gillespie, C. S. (2022). "Bayesian experimental design without posterior calculations: an adversarial approach. Supplementary material." *Bayesian Analysis*. doi: https://doi.org/10.1214/22-BA1306SUPP. 135

Price, D. J., Bean, N. G., Ross, J. V., and Tuke, J. (2018). An induced natural selection heuristic for finding optimal Bayesian experimental designs. *Computational Statistics & Data Analysis*, 126:112–124. MR3808393. doi: https://doi.org/10.1016/j.csda.2018.04.011. 136

Pronzato, L. and Walter, É. (1985). Robust experiment design via stochastic approximation. *Mathematical Biosciences*, 75(1):103–120. MR0800967. doi: https://doi.org/10.1016/0025-5564(85)90068-9. 136, 145

Ryan, E. G., Drovandi, C. C., McGree, J. M., and Pettitt, A. N. (2016). A review of modern computational algorithms for Bayesian optimal design. *International Statistical Review*, 84(1):128–154. MR3491282. doi: https://doi.org/10.1111/insr.12107. 138, 139, 141, 145

Ryan, E. G., Drovandi, C. C., Thompson, M. H., and Pettitt, A. N. (2014). Towards Bayesian experimental design for nonlinear models that require a large number of sampling times. *Computational Statistics & Data Analysis*, 70:45–60. MR3125477. doi: https://doi.org/10.1016/j.csda.2013.08.017. 133, 136, 151, 156

Shao, S., Jacob, P. E., Ding, J., and Tarokh, V. (2019). Bayesian model comparison with the Hyvärinen score: computation and consistency. *Journal of the American Statistical Association*, 114(528):1–24. MR4047303. doi: https://doi.org/10.1080/01621459.2018.1518237. 158

Van der Vaart, A. W. (2000). *Asymptotic statistics*. Cambridge university press. MR1652247. doi: https://doi.org/10.1017/CBO9780511802256. 142

Walker, S. G. (2016). Bayesian information in an experiment and the Fisher information distance. *Statistics & Probability Letters*, 112:5–9. MR3475481. doi: https://doi.org/10.1016/j.spl.2016.01.014. 133, 134, 135, 139, 141, 158

Wolfson, L. J., Kadane, J. B., and Small, M. J. (1996). Expected utility as a policy-making tool: an environmental health example. *Statistics Textbooks and Monographs*, 151:261–278. 138

Acknowledgments

The authors thank Kevin Wilson, Chris Oates, John Matthews, Phil Dawid, Kyle Cranmer, Iain Murray and journal referees/editors for helpful comments, and gratefully acknowledge an EPSRC studentship supporting Sophie Harbisher.